

## KNOWLEDGE SPACES AND THE COMPLETENESS OF LEARNING STRATEGIES

STEFANO BERARDI<sup>a</sup> AND UGO DE'LIGUORO<sup>b</sup>

<sup>a,b</sup> Dipartimento di Informatica, Università di Torino c.so Svizzera 185 Torino, Italy  
*e-mail address:* {stefano,deligu}@di.unito.it

**ABSTRACT.** We propose a theory of learning aimed to formalize some ideas underlying Coquand's game semantics and Krivine's realizability of classical logic. We introduce a notion of knowledge state together with a new topology, capturing finite positive and negative information that guides a learning strategy. We use a leading example to illustrate how non-constructive proofs lead to continuous and effective learning strategies over knowledge spaces, and prove that our learning semantics is sound and complete w.r.t. classical truth, as it is the case for Coquand's and Krivine's approaches.

### INTRODUCTION

Several methods have been proposed to give a recursive interpretation of non-recursive constructions of mathematical objects, whose existence and properties are classically provable. A non-exhaustive list includes the continuation-based approach initiated by Griffin [16], the game theoretic semantics of classical arithmetic by Coquand [10] and Krivine's realizability of classical logic [18].

As observed by Coquand, there is a common informal idea underlying the different approaches, which is learning. With respect to the dialogic approach, learning consists into interpreting the strategy for the defender of a statement against the refuter by a strategy guiding the interaction between a learning agent and the “world”, representing what can be experienced by direct computation.

Under the influence of Gold's ideas [14, 15] and of Hayashi's Limit Computable Mathematics [17] we have proposed a formal theory of “learning” and “well founded limits” in [3]. In the theory the goal of the learning process is to find an evidence, or a witness as it is usually called, of the truth of some given sentence, which is the “problem” that the learning strategy solves. Such an evidence is always tentative, since certainty can be attained only

---

*2012 ACM CCS:* [Theory of computation]: Models of computation—Interactive computation; Logic—Constructive mathematics.

*Key words and phrases:* Classical Logic, Proof Mining, Game Semantics, Learning, Realizability.

<sup>a,b</sup> This work was partially supported by PRIN project n. 2008H49TEH.

in the ideal limit. The task of the learning strategy is to tell how to react to the discovery that the current guess is actually wrong, and this is done on the basis of the knowledge collected in the learning process, which includes all the “counterexamples” that have been seen up to the time.

Here we propose the same idea, but in the different perspective of topological spaces and continuous maps. We assume having an ideal object being the result of some non-effective mental construction and satisfying some decidable property. We say that this construction may be learned w.r.t. the property if we can find a “finite approximation” of the construction which still satisfies the property. In particular given a classical proof of an existential statement, we see the computational content of the proof as the activity of guessing more and more about the object (individual) the statement is about, without ever obtaining a full information; in a discrete setting such as natural numbers, the approximation is actually a different object, which we think as “close” to the ideal one, the “limit”, only with respect to some given property which both satisfy.

When reasoning about ideal objects, we deal with descriptions rather than with the objects themselves. While learning of an ideal object we have step by step certain amounts of knowledge, which consist of pieces of evidence (e.g. decidable statements): therefore we topologize *states of knowledge* to express the idea that a continuous strategy only depends on finite positive and negative information to yield finite approximations of the ideal limit. We call *interactive realizer* any continuous function of states of knowledge that roughly tells which are the further guesses to improve the given knowledge, and how to react to the discovery of negative evidences (e.g. counterexamples to certain assumptions). We conjecture that interactive realizers can be seen as winning strategies in the sense of Coquand and as the semantics of lambda terms with continuations in Griffin’s work and of terms in a call-by-value version of Krivine’s classical realizability interpretation; however this far from being evident and it deserves further investigation.

We call a *model* any perfect (usually infinite) knowledge state. The main result of this paper is that any object that can be ideally learned in a model can be effectively learned in a finite state of knowledge approximating the model, and this state is found by means of a realizer. Since models represent classical truth, the completeness theorem can be read as stating that the learning semantics for classical proofs is complete w.r.t. classical truth, namely Tarskian truth, as it is the case for Coquand’s dialogic semantics of proofs and for Krivine’s classical realizability, and that the learning process is indeed effective.

The paper is organized as follows. In §1 we introduce a motivating example, which is used throughout the paper. In §2 we define the structures of states of knowledge and their topology. In §3 the concept of relative truth is introduced to define sound, complete and model knowledge states. In §4 we define interactive realizability and prove the completeness theorem. Finally in §5 we show, although not in full generality, how the existence of a computable realizer over a computable space of knowledge gives rise to an algorithm to effectively learn approximated solutions.

**Related works.** The suggestion by Coquand that the dialogic interpretation of classical proofs could be seen as learning of some abstract entities can be found in [9], a preliminary version of [10]. The idea has been illustrated by means of a suggestive analysis of the proof of Sylvester’s Conjecture by von Plato in [20]. Beside Krivine’s [18], Miquel’s work in [19] illustrates in detail the behaviour of classical realizability of existential statements (also

in comparison to Friedman’s method), and we strongly believe that the construction is a learning process in the sense of the present paper.

Learning in the limit of undecidable properties and ideal entities comes from Gold’s work [14, 15], and has been recently formulated for classical logic by Hayashi e.g. in [17]. Hayashi defined realizers as limits over a linearly ordered knowledge set and in the case of monotonic learning: the first model with realistic assumptions over of the knowledge states, again in the case of monotonic learning, is in [2]. We have investigated more the concept of learning in the limit incorporating Coquand’s ideas in [3], although in a combinatory rather than topological perspective. We have further elaborated the concept of learnable in the limit in [4], where a “solution” in terms of the following §4 is called “individual”, since we identify the ideal limit with the map reading the learned values out states of knowledge. The formal definition of the state topology, however, is new with the present work, as well as the treatment of the general, non-monotonic case. Also the concept of “interactive realizability” has been introduced in [4], but in the simpler case of monotonic learning. Essentially the same construction as in [4] is used in [1] to define a realizability interpretation of **HA** plus excluded middle restricted to  $\Sigma_1^0$ -formulas. It turns out that interactive realizability is a generalization of Kleene’s realizability, and this motivates the terminology. A geometrical application of interactive realizability, akin to Von Plato’s work on Sylvester’s Conjecture, can be found in Birollo’s ph.d. thesis [8]. Recently Berardi and Steila analyzed Erdős’ proof of Ramsey Theorem [7], where it is shown that there exists an interpretation of such a proof using a three-level non-monotonic learning structure.

## 1. SOLVING PROBLEMS BY LEARNING

To illustrate the idea of learning strategies, either monotonic or non-monotonic, we propose an example suggested by Coquand and developed by Fridlender in [11]. Let  $f_1, f_2$  be total functions over  $\mathbb{N}$ . Fix some integer  $k > 0$  and consider the statements: “there is an increasing sequence of  $k$  integers which is weakly increasing w.r.t.  $f_1$ ” and “there is an increasing sequence of  $k$  integers which is weakly increasing w.r.t.  $f_1$  and  $f_2$ ”. Formally:

$$\exists x_1 \dots \exists x_k. x_1 < \dots < x_k \wedge f_1(x_1) \leq \dots \leq f_1(x_k) \quad (1.1)$$

$$\exists x_1 \dots \exists x_k. x_1 < \dots < x_k \wedge f_1(x_1) \leq \dots \leq f_1(x_k) \wedge f_2(x_1) \leq \dots \leq f_2(x_k) \quad (1.2)$$

We look at these statements as the *problems* of finding a  $k$ -tuple  $n_1 < \dots < n_k$  witnessing their truth. We begin by observing that these statements can be proved classically as follows. For any  $f : \mathbb{N} \rightarrow \mathbb{N}$  and  $A \subseteq \mathbb{N}$  we say that “ $n$  is a local minimum of  $f$  w.r.t.  $A$ ” (shortly  $n$  is an  $f, A$ -minimum) if  $n$  is the minimum of  $f$  in  $A \cap [n, \infty[$ . Formally:

$$f, A\text{-min}(n) \Leftrightarrow \forall y \in A. n < y \Rightarrow f(n) \leq f(y).$$

Observe that the predicate  $f, A\text{-min}(n)$  is undecidable in general, even if  $f$  is recursive and  $A$  decidable. The statement  $\neg f, A\text{-min}(n)$  is classically equivalent to  $\exists y \in A. n < y \wedge f(n) > f(y)$ . For any  $f : \mathbb{N} \rightarrow \mathbb{N}$  and infinite  $A \subseteq \mathbb{N}$ , we denote by

$$A_f = \{a \in A \mid f, A\text{-min}(a)\} \subseteq A$$

the set of all  $f, A$ -minima. We now study the set  $A_f$ .

**Lemma 1.1.** *For any  $f : \mathbb{N} \rightarrow \mathbb{N}$  and infinite  $A \subseteq \mathbb{N}$ , the set  $A_f = \{a \in A \mid f, A\text{-min}(a)\}$  of  $f, A$ -minima is infinite, and  $f$  is monotonic over  $A_f$ .*

*Proof.* Toward a contradiction suppose that  $A_f$  is finite, possibly empty. Then there exists  $a_0 = \min(A \setminus A_f)$ , as  $A$  is infinite. By definition of  $a_0$  for all  $a \in A$  with  $a \geq a_0$  we have  $\neg f, A\text{-min}(a)$ , that is: there is some  $a' > a$ ,  $a' \in A$  such that  $f(a) > f(a')$ . If we choose  $a = a_0$  we deduce that there exists some  $a_1 \in A$  such that  $a_0 < a_1$  and  $f(a_0) > f(a_1)$ , and so on. By iterating the reasoning we get an infinite sequence  $a_0 < a_1 < a_2 < \dots$  such that  $f(a_i) > f(a_{i+1})$  for all  $i \in \mathbb{N}$ , which is on turn an infinite descending chain in  $\mathbb{N}$ , that is an absurdity.  $\square$

**Theorem 1.2.** *Both statements (1.1) and (1.2) are classically provable.*

*Proof.* In Lemma 1.1 take  $A = \mathbb{N}$ : then  $f_1$  is monotonic over the infinite set  $A_{f_1}$ . If we take the first  $k$  elements of  $A_{f_1}$  we have an increasing sequence of  $k$  integers whose values are weakly increasing under  $f_1$ , establishing (1.1).

To prove (1.2) we use again Lemma 1.1.2 taking  $A = \mathbb{N}_{f_1}$ , which we know to be infinite by the same lemma; then  $f_1, f_2$  are both monotonic over the infinite set  $A_{f_2} = (\mathbb{N}_{f_1})_{f_2}$ .  $\square$

The proof of Lemma 1.1 and its use in the proof of Theorem 1.2 are non-constructive, as they rely on the “computation” of the minimum of  $f$  in  $A$  for certain  $f$  and  $A$ . In order to compute the minimum of  $f$  we need, in general, to know infinitely many values of  $f$ . However, this proof may be interpreted by a computation as soon as we only require finitely many  $n_1, \dots, n_k$  that satisfy (1.1) or (1.2). This is because  $n_1, \dots, n_k$  can be found using a finite knowledge about  $f$ .

Indeed the proof of Lemma 1.1 can be turned into an effective strategy to learn a solution to problem (1.1), assuming that  $f_1$  is recursive. The basic remark is that we do not actually need to know the infinitely many elements of  $\mathbb{N}_{f_1}$ , nor we have to produce some  $n$  which belongs to  $\mathbb{N}_{f_1}$  *beyond any doubt*. We can approximate the infinite set  $\mathbb{N}_{f_1}$  by some finite set  $B$ , whose elements are not necessarily in  $\mathbb{N}_{f_1}$ , rather they are just  $f_1, \mathbb{N}$ -minima *as far as we know*.  $B$  is a kind of hypothesis about  $\mathbb{N}_{f_1}$ .

More precisely, we will find a set  $B$  such that  $f_1, B\text{-min}(n)$  for all  $n \in B$ . This property is trivially true for any singleton set, say  $\{0\}$ . In the general case, if  $B$  has  $k$  elements or more, then by definition of  $f_1, B\text{-min}$  the set  $B$  is a solution to (1.1). Otherwise we take any  $m > \max(B)$  and try adding  $m$  to  $B$ . Since we cannot decide whether any  $n$  is a local minimum of  $f_1$ , we are not allowed to increase  $B$  to  $B \cup \{m\}$ , because it could be the case that  $f_1(n) > f_1(m)$  for some  $n \in B$ . Rather we update  $B$ , by removing all  $n \in B$  such that  $f_1(n) > f_1(m)$ :

$$B' = \{n \in B \mid f_1(n) \leq f_1(m)\} \cup \{m\},$$

The new set  $B'$  includes  $m$  and satisfies the invariant property of containing only  $f_1, B'$ -minima. The cardinality of  $B'$  is not necessarily greater than that of  $B$ , so that we need an argument to conclude that, starting from the singleton  $\{0\}$  and iterating the step from  $B$  to  $B'$ , the learning agent will eventually reach a  $k$ -element set with the required property.

The termination argument in this case works as follows. Although the sequence of sets is not increasing w.r.t. inclusion, the knowledge that some elements are *not* local minima of  $f_1$  grows monotonically, since more and more pairs  $n, m$  are found such that  $f_1(n) > f_1(m)$ . From this remark, one can prove by a fixed-point argument (over a suitable topology) that the growth of knowledge eventually ends, which implies that a set  $B$  with  $k$  elements will be found after finitely many steps. In this case we speak of *monotonic learning*.

In order to include an example of non-monotonic learning, we assume that both  $f_1$  and  $f_2$  are recursive, and we outline an effective computation approximating an initial segment of  $(\mathbb{N}_{f_1})_{f_2}$ , solving problem (1.2). The informal interpretation we include here will be formalized in the following sections using the notion of layered valuation.

As it happens in the classical proof of Theorem 1.2, we iterate the same method used for problem (1.1), and build a  $C \subseteq B$  of  $f_2$ ,  $C$ -minima, where  $B$  is the current approximation of the infinite set  $\mathbb{N}_{f_1}$ . In doing so the learning agent assumes that  $B$  is a subset of  $\mathbb{N}_{f_1}$  (though she cannot be certain of this), and that all elements of  $C$  are  $f_2$ -minima w.r.t.  $\mathbb{N}_{f_1}$  (again an uncertain belief). At each step, the learner takes some  $m \in B$ , such that  $m > \max(C)$ , and manages to add  $m$  to  $C$ , possibly by removing some of its elements, by computing:

$$C' = \{p \in C \mid f_2(p) \leq f_2(m)\} \cup \{m\} \subseteq B.$$

This is only possible when such an  $m$  exists in  $B$ : if not the algorithm generating  $B$  has to be resumed to get a larger set containing an element greater than  $\max(C)$ . But since  $B$  does not grow monotonically, elements of  $C$  will be dropped while computing  $C'$  also because they are no longer in  $B$ . This makes the convergence proof much harder. Indeed the knowledge accumulated while building  $B$  takes the simple form of (sets of statements)  $f_1(n) > f_1(m)$ , and it grows monotonically; on the contrary the “knowledge” gathered while computing  $C$  consists of more complex statements of the form:  $m \in B \wedge f_2(n) > f_2(m)$ , with  $B$  changing non-monotonically during the computation of  $C$ . This knowledge is the conjunction of the hypothesis (uncertain beliefs)  $m \in B$  and the fact  $f_2(n) > f_2(m)$  (a decidable statement). This second layer of knowledge, mixing hypothesis and facts, does not grow monotonically, because any hypothesis  $m \in B$  may turn out to be false: therefore it is unsafe, yet it guides the construction of  $C$ . In this case we speak of *non-monotonic learning*. Non-monotonic learning is the most general form of learning.

## 2. STATES OF KNOWLEDGE AND THEIR TOPOLOGY

There are three kinds of entities in learning: questions, answers and states of knowledge. The main concern are states of knowledge, which on turn are certain sets of answers. Answers are viewed as atomic objects although in the examples they are logical formulas, since their internal structure is immaterial from the point of view of the topology of states of knowledge we are introducing here, while their complexity is abstractly represented by levels in the next section. Questions instead are represented indirectly by equivalence classes of answers, each to be thought of as the set of alternative, incompatible choices for an answer to the same question.

**Definition 2.1** (Knowledge Structure and State of Knowledge). *A knowledge structure  $(\mathbb{A}, \sim)$  consists of a non-empty countable set  $\mathbb{A}$  of answers and an equivalence relation  $\sim \subseteq \mathbb{A} \times \mathbb{A}$ . As a topological space,  $\mathbb{A}$  is equipped with the discrete topology.*

*The set  $\mathbb{Q} = \mathbb{A}/\sim$  of equivalence classes  $[x]$  w.r.t.  $\sim$  is the set of questions, and it is equipped with the discrete topology.*

*A subset  $X \subseteq \mathbb{A}$  is a state of knowledge if for all  $x \in \mathbb{A}$  the set  $X \cap [x]$  is either empty or a singleton. We denote by  $\mathbb{S}$  the set of knowledge states and by  $\mathbb{S}_{\text{fin}}$  the subset of finite elements of  $\mathbb{S}$ .*

If  $x \sim y$  then  $x, y$  are two answers to the same question. The equivalence class  $[x]$  abstractly represents the question answered by  $x$ . Two answers  $x, y \in \mathbb{A}$  are *compatible*,

written  $x\#y$ , if they are not different answers to the same question, namely if:

$$x \sim y \Rightarrow x = y.$$

Now it turns out that  $\mathbb{S}$  is the coherence space whose web is the graph  $(\mathbb{A}, \#)$  (see [13] §8.2). Indeed we observe that  $\mathbb{S}$  is a poset by subset inclusion, and it is downward closed. It follows that  $(\mathbb{S}, \cap, \subseteq)$  is an inf-semilattice with bottom  $\emptyset$ .  $\mathbb{S}$  is closed under arbitrary but non-empty inf, as the empty inf, namely the whole set  $\mathbb{A}$ , is not consistent in general. Indeed  $\mathbb{S}$  is not closed under union, unless the compatibility relation is the identity. We say that  $X$  and  $Y$  are compatible w.r.t. inclusion, if  $X \subseteq Z \supseteq Y$  for some  $Z \in \mathbb{S}$ . Clearly the union of a family  $\mathcal{U} \subseteq \mathbb{S}$  belongs to  $\mathbb{S}$  if and only if all elements of  $\mathcal{U}$  are pairwise compatible sets. By this  $\mathbb{S}$  is closed under directed and in general bounded sups, so it is a domain which in fact has compacts  $K(\mathbb{S}) = \mathbb{S}_{fin}$  and is algebraic.

**Example 1** Let us reconsider the example in §1. A knowledge structure  $(\mathbb{A}_0, \sim_0)$  for learning a solution to (1.1) can be defined by taking  $\mathbb{A}_0 = \{(n, m) \in \mathbb{N} \times \mathbb{N} \mid n < m\}$ , where we interpret a pair  $(n, m)$  as the statement: “ $m$  is a counterexample to  $f_1, \mathbb{N}\text{-min}(n)$ ”, and more precisely as the formula:

$$n < m \wedge f_1(n) > f_1(m).$$

If we think of  $m$  as the answer to the question about  $n$ , we obtain the definition of the relation  $(n, m) \sim_0 (n', m')$  by  $n = n'$ .

A knowledge structure  $(\mathbb{A}_1, \sim_1)$  for learning a solution to (1.2) can be defined by taking  $\mathbb{A}_1 = \mathbb{A}_0$  and  $\sim_1 = \sim_0$ . However  $(\mathbb{A}_1, \sim_1)$  has not the same intended meaning as  $(\mathbb{A}_0, \sim_0)$ ; indeed an answer  $(n, m) \in \mathbb{A}_1$  is interpreted by the statement:

$$n < m \wedge f_2(n) > f_2(m) \wedge n, m \in \mathbb{N}_{f_1}.$$

Finally we set  $\mathbb{A}_2 = \mathbb{A}_0 \uplus \mathbb{A}_1 = \{(i, n, m) \mid i \in \{0, 1\} \ \& \ (n, m) \in \mathbb{A}_i\}$ , namely the disjoint union of  $\mathbb{A}_0$  and  $\mathbb{A}_1$  and we define  $(i, n, m) \sim_2 (j, n', m')$  if and only if  $i = j$  and  $n = n'$  (that is  $(n, m) \sim_i (n', m')$ ).

Let  $\mathbb{S}_2$  be the knowledge space associated to  $\mathbb{A}_2$  and  $X \in \mathbb{S}_2$  be any state of knowledge. Then we interpret  $(0, n, m) \in X$  by “the agent knows at  $X$  that  $n < m$  and  $f_1(n) > f_1(m)$ ”, hence that  $n$  is not in  $\mathbb{N}_{f_1}$ . We interpret: for all  $p \in \mathbb{N}$ ,  $(0, n, p) \notin X$  by “the agent knows at  $X$  of no  $p$  such that  $n < p$  and  $f_1(n) > f_1(p)$ ”, hence she believes that  $n$  is in  $\mathbb{N}_{f_1}$ . We write  $(\mathbb{N}_{f_1})^X = \{n \in \mathbb{N} \mid \forall p \in \mathbb{N}. (0, n, p) \notin X\}$  for the set of  $n$  which the agent believes to be in  $\mathbb{N}_{f_1}$  at  $X$ . In the same way we write  $((\mathbb{N}_{f_1})_{f_2})^X = \{n \in (\mathbb{N}_{f_1})^X \mid \forall p \in \mathbb{N}. (1, n, p) \notin X\}$  for the set of  $n$  which the agent believes to be in  $(\mathbb{N}_{f_1})_{f_2}$  at  $X$ . We will see in the following sections that  $(\mathbb{A}_2, \sim_2)$  is a knowledge structure apt to learn (1.2).

In a state of knowledge  $X$  two distinct answers  $x, y \in X$  must be compatible, that is  $x \not\sim y$ . We consider two answers to the same question as incompatible because we see them as the outcome of a computation process, that returns at most one answer to each question. In this perspective a question can be figured as a memory cell, whose value is either undefined or just one, and that can change in time. A knowledge state is an abstraction of the state of the whole memory, hence recording only compatible answers.

The “state of knowledge” of a finite agent should be finite; for the sake of the theory we also consider infinite states of knowledge, which are naturally approximated by finite ones in a sense to be made precise by a topology. Beside these conceptual reasons, the topological treatment that follows will be essential in the proofs of Theorems 4.7 and 4.8,

as well as in the proof of Theorem 5.6 establishing convergence of the algorithm in section 5.

Let us define a query map  $q : \mathbb{S} \times \mathbb{Q} \rightarrow \mathcal{P}_{fin}(\mathbb{A})$ , where  $\mathcal{P}_{fin}(\mathbb{A})$  is the set of finite subsets of  $\mathbb{A}$ , by  $q(X, [x]) = X \cap [x]$ . Then  $q(X, [x])$  is either a singleton  $\{y\}$ , meaning that, at  $X$ ,  $y$  is the answer to the question  $[x]$ , or the empty set, meaning that the agent knows at  $X$  of no answer to  $[x]$ , and she assumes that there is none. Take the discrete topology over  $\mathcal{P}_{fin}(\mathbb{A})$ ; we then consider the smallest topology over  $\mathbb{S}$  making  $q$  continuous.

**Definition 2.2** (State Topology). *The state topology  $(\mathbb{S}, \Omega(\mathbb{S}))$  is generated by the subbasics  $A_x, B_x$ , with  $x \in \mathbb{A}$ :*

$$\begin{aligned} A_x &= \{X \in \mathbb{S} \mid x \in X\} = \{X \in \mathbb{S} \mid X \cap [x] = \{x\}\}, \\ B_x &= \{X \in \mathbb{S} \mid X \cap [x] = \emptyset\}. \end{aligned}$$

Let  $X, Y, Z$  range over  $\mathbb{S}$ , and  $s, t$  over  $\mathbb{S}_{fin} = \mathbb{S} \cap \mathcal{P}_{fin}(\mathbb{A})$ . By definition, a basic open of  $\Omega(\mathbb{S})$  has the shape:

$$\mathcal{O}_{U,V} = \bigcap_{x \in U} A_x \cap \bigcap_{y \in V} B_y,$$

for finite  $U, V \subseteq \mathbb{A}$ . If  $\neg x \# y$ , that is  $x \sim y$  and  $x \neq y$ , then  $A_x \cap A_y = \emptyset$ , because if  $x, y \in X$  then  $X$  would be inconsistent, so that  $\emptyset$  is a basic open. On the other hand if  $x \sim y$  then  $B_x = B_y$ . Therefore without loss of generality we assume  $U, V$  to be consistent, and that all basic opens of  $\Omega(\mathbb{S})$  are of the form  $\mathcal{O}_{U,V} = \mathcal{O}_{s,t}$ , for some  $s, t \in \mathbb{S}_{fin}$ . Summing up, assume that  $s = \{x_1, \dots, x_n\}$  and  $t = \{y_1, \dots, y_m\}$ . Then  $X \in \mathcal{O}_{s,t}$  means that at  $X$  the agent knows the answers  $x_1, \dots, x_n$  to the questions  $[x_1], \dots, [x_n]$  (a finite positive information), while she knows of no answer to the questions  $[y_1], \dots, [y_m]$ , and assumes that there is none (a finite negative information).

The same thing can be looked at by observing that the continuous maps  $f : \mathbb{S} \rightarrow I$ , with  $I$  a discrete space, are those “asking finitely many questions to their argument”. By this we mean:  $f$  is continuous if and only if for all  $X \in \mathbb{S}$  there are finitely many questions  $[x_1], \dots, [x_n] \in \mathbb{Q}$  such that for all  $Y \in \mathbb{S}$ , if  $q(X, [x_i]) = q(Y, [x_i])$  for  $i = 1, \dots, n$  then  $f(X) = f(Y)$ .

The state topology is distinct from, yet strictly related to, several well-known topologies.  $\Omega(\mathbb{S})$  is discrete if and only if  $\mathbb{Q}$  is finite (there are finitely many equivalence classes).  $\Omega(\mathbb{S})$  is homeomorphic to the product space  $\prod_{[x] \in \mathbb{Q}} ([x] \uplus 1)$  of the discrete topologies over  $[x] \uplus 1$ , where 1 is any singleton (representing the “undefined” answer to the question  $[x]$ ) and  $\uplus$  is disjoint union. Every state topology is homeomorphic to some subspace of the Baire topology over  $\mathbb{N}^{\mathbb{N}}$ . The state topology is totally disconnected and Hausdorff: it is compact (namely it is a Stone space) if and only if all equivalence classes are finite. If all equivalence classes in  $\mathbb{Q}$  are singletons (that is, if  $\sim$  is the equality relation on  $\mathbb{A}$ ) and  $\mathbb{Q}$  is infinite then  $\Omega(\mathbb{S})$  is homeomorphic to the Cantor space  $2^{\mathbb{N}}$ . If all equivalence classes in  $\mathbb{Q}$  are infinite and  $\mathbb{Q}$  is infinite, then  $\Omega(\mathbb{S})$  is homeomorphic to the whole Baire space.

A clopen is an open and closed set; hence clopens are closed under complement, finite unions and intersections. There are significative examples of clopen sets in  $\mathbb{S}$ .

**Lemma 2.3** (Subbasic opens of State Topology are clopen). *Assume that  $x \in \mathbb{A}$ ,  $f : \mathbb{S} \rightarrow I$  is continuous,  $I$  is a discrete space and  $J \subseteq I$ . Then:*

- (1)  $B_x$  is clopen
- (2)  $A_x$  is clopen
- (3)  $f^{-1}(J)$  is clopen.

*Proof.*

- (1) It is enough to prove that  $\mathbb{S} \setminus B_x$  is equal to the union  $\bigcup \{A_y \mid y \in [x]\}$  of open sets. Let  $X \in \mathbb{S}$  and  $x \in \mathbb{A}$ , then:

$$\begin{aligned} X \notin B_x &\Leftrightarrow X \cap [x] \neq \emptyset \\ &\Leftrightarrow \exists y \in X. x \sim y \\ &\Leftrightarrow \exists y. y \in [x] \ \& \ X \in A_y \end{aligned}$$

- (2)  $\mathbb{S} \setminus A_x$  is some union of open sets. Indeed we have

$$X \notin A_x \Leftrightarrow X \in B_x \cup \bigcup \{A_y \mid y \in [x] \ \& \ y \neq x\}.$$

- (3) We have  $\mathbb{S} \setminus f^{-1}(J) = f^{-1}(I \setminus J)$ . Both  $f^{-1}(J), f^{-1}(I \setminus J)$  are open because they are the counter-image under some continuous  $f$  of  $J, (I \setminus J)$ , which are open sets because  $I$  is discrete.  $\square$

As a consequence of Lemma 2.3.1 we have that the predicate  $n \in \mathbb{N}_{f_1}^X$  is continuous in  $X$ , since  $\{X \mid n \in \mathbb{N}_{f_1}^X\} = B_{(0,n,n+1)}$  which is a clopen set, and indeed  $n \notin \mathbb{N}_{f_1}^X$  if and only if  $X \in \bigcup_{m>n} A_{(0,n,m)}$ , namely the complement of  $B_{(0,n,n+1)}$ . A similar remark holds for  $n \in (\mathbb{N}_{f_1})_{f_2}^X$ .

It is instructive to compare the state topology to Scott and Lawson topologies over  $\mathbb{S}$ . The *Scott topology* over the cpo  $(\mathbb{S}, \subseteq)$  is determined by taking all the  $A_x$  with  $x \in \mathbb{A}$  as subbasics. On the other hand any  $B_x$  is a Scott-closed set, since its complement  $\mathbb{S} \setminus B_x$  is equal to the union  $\bigcup \{A_y \mid y \in [x]\}$  of Scott opens. However  $B_x$  is not Scott-open because it is not upward closed.

Recall that (see [12]) the *lower topology* over a poset is generated by the complements of principal filters; the *Lawson topology* is the smallest refinement of both the lower and the Scott topology. In case of the cpo  $(\mathbb{S}, \subseteq)$  the Lawson topology is generated by the subbasics:

$$\overline{X \uparrow} = \{Y \in \mathbb{S} \mid X \not\subseteq Y\} \text{ and } s \uparrow = \{Y \in \mathbb{S} \mid s \subseteq Y\},$$

for  $X \in \mathbb{S}$  and  $s \in \mathbb{S}_{fin}$ , representing the negative and positive information respectively.

The state topology includes the Lawson topology, and in general it is finer than that. The next lemma tells that all Lawson opens are open in the state topology, but if some equivalence class  $[x]$  is infinite then some open of the state topology is not open in the Lawson topology. Recall that  $\Omega(\mathbb{S})$  denotes the family of open sets in the state topology.

**Lemma 2.4.**

- (1) *All basic opens of Lawson topology are in  $\Omega(\mathbb{S})$ .*  
(2) *For all  $x \in \mathbb{A}$ ,  $B_x$  is Lawson-open if and only if  $[x]$  is finite.*

*Proof.*

- (1) Since  $X \uparrow = \bigcap_{x \in X} x \uparrow$ , writing  $x \uparrow$  for  $\{x\} \uparrow$ , we have  $\overline{X \uparrow} = \bigcup_{x \in X} \overline{x \uparrow}$ , hence it suffices to prove that  $\overline{x \uparrow} \in \Omega(\mathbb{S})$ , for any  $x \in \mathbb{A}$ ; since  $\overline{x \uparrow} = \{Y \in \mathbb{S} \mid x \not\subseteq Y\}$ , this is established by the equality:

$$\begin{aligned} \overline{x \uparrow} &= \{Y \in \mathbb{S} \mid Y \cap [x] = \emptyset\} \cup \bigcup_{x \# y} \{Z \in \mathbb{S} \mid Z \cap [x] = \{y\}\} \\ &= B_x \cup \bigcup_{x \# y} A_y \end{aligned}$$

where  $x \# y$  is equivalent to  $x \neq y$  and  $[x] = [y]$ .



- (2) Observe that  $B_x = \bigcap_{y \in [x]} \overline{y \uparrow}$ ; therefore if  $[x] = \{x_1, \dots, x_k\}$  then  $B_x = \overline{x_1 \uparrow} \cap \dots \cap \overline{x_k \uparrow}$  is a finite intersection of open sets in the Lower topology, hence it is open w.r.t. the Lawson topology. Vice versa suppose that  $[x]$  is infinite. Toward a contradiction let

$$\emptyset \neq \mathcal{O} = \overline{X_1 \uparrow} \cap \dots \cap \overline{X_k \uparrow} \cap s_1 \uparrow \cap \dots \cap s_h \uparrow \subseteq B_x$$

be a basic non empty open of the Lawson topology. If we take any  $Y \in \mathcal{O}$ , then by definition we have  $\{x_1, \dots, x_k\} \cap Y = \emptyset$  for some  $x_1 \in X_1, \dots, x_k \in X_k$ , and  $U = s_1 \cup \dots \cup s_h \subseteq Y$  is finite consistent. By the hypothesis that  $[x]$  is infinite there is some  $y \in [x] \setminus \{x_1, \dots, x_k\}$ . By the assumption that  $U \subseteq Y \in \mathcal{O} \subseteq B_x$ , we have that  $U \cap [x] \subseteq Y \cap [x] = \emptyset$ , hence  $V = U \cup \{y\} \in \mathbb{S}_{fin}$ , namely  $V$  is consistent. Now:

$$V \cap \{x_1, \dots, x_k\} = (U \cup \{y\}) \cap \{x_1, \dots, x_k\} \subseteq (Y \cup \{y\}) \cap \{x_1, \dots, x_k\} = \emptyset$$

From  $U \subseteq V$  and  $V \cap \{x_1, \dots, x_k\} = \emptyset$  we deduce that  $V \in \mathcal{O}$ , but  $V \notin B_x$  since  $y \in V \cap [x]$ : a contradiction.  $\square$

As an immediate consequence of Lemma 2.4 we have the following.

**Theorem 2.5** (State versus Lawson Topology). *The state topology  $\Omega(\mathbb{S})$  refines the Lawson topology over the cpo  $(\mathbb{S}, \subseteq)$ , and they coincide if and only if  $[x]$  is finite for all  $x \in \mathbb{A}$ .*

### 3. RELATIVE TRUTH AND LAYERED STATES

Answers to a question can be either true or false. In the perspective of learning we think of truth values with respect to the actual knowledge that a learning agent can have at some stage of the process, so that we relativize the valuation of the answers to the knowledge states. Furthermore the example of learning the solution to problem (1.2) in §1 shows that there can be dependencies among answers in a state of knowledge. We formalize this by means of the stratification into levels of the set of answers. In the example of §1 we only need levels 0 and 1. In the definition, however, we allow any number of levels, even transfinite.

Denote with  $Ord$  the class of denumerable ordinals. Let us assume the existence of a map  $\text{lev} : \mathbb{A} \rightarrow Ord$ , associating to each answer  $x$  the ordinal  $\text{lev}(x)$ , and such that any two answers to the same question are of the same level. Thus, implicitly, the level is assigned to the questions. If  $X \in \mathbb{S}$  and  $\alpha \in Ord$  we write  $X \upharpoonright \alpha = \{x \in X \mid \text{lev}(x) < \alpha\}$ . We can now make precise the notions of level and of truth of an answer w.r.t. a knowledge state. Let us denote with  $2 = \{\text{true}, \text{false}\}$  the set of truth values.

**Definition 3.1** (Layered Knowledge Structures). *Let  $(\mathbb{A}, \sim)$  be a knowledge structure. Then a layered knowledge structure is a tuple  $(\mathbb{A}, \sim, \text{lev}, \text{tr})$  where  $\text{lev} : \mathbb{A} \rightarrow Ord$  and  $\text{tr} : \mathbb{A} \times \mathbb{S} \rightarrow 2$  are two maps such that:*

- (1) Two answers to the same question have the same level:

$$\forall x, y \in \mathbb{A}. (x \sim y \Rightarrow \text{lev}(x) = \text{lev}(y))$$

- (2)  $\text{tr}$  is continuous, by taking  $\mathbb{A}$  and  $2$  with the discrete topology,  $\mathbb{S}$  with the state topology  $\Omega(\mathbb{S})$ , and  $\mathbb{A} \times \mathbb{S}$  with the product topology;

- (3)  $\text{tr}$  is layered:  $\forall x \in \mathbb{A}, X \in \mathbb{S}. \text{tr}(x, X) = \text{tr}(x, X \upharpoonright \text{lev}(x))$

The ordinal  $\text{lev}(x)$  represents the reliability of the answer  $x$ , which is maximal when  $\text{lev}(x) = 0$ ; by this reason we ask that all the answers to the same question have the same level. We prefer to assign levels to answers instead of questions because the formers are the only concrete objects of a knowledge structure.

The mapping  $\text{tr}$  of a layered knowledge structure is a boolean valuation of atoms that continuously depends on states of knowledge; we call it a *layered valuation* because the value of an atom w.r.t. a state (to which it does not necessarily belong) only depends on the atoms of lower level belonging to that state. Set  $\mathbf{T}_x = \text{tr}(x)^{-1}(\{\text{true}\}) = \{X \in \mathbb{S} \mid \text{tr}(x, X) = \text{true}\}$ , and similarly  $\mathbf{F}_x = \text{tr}(x)^{-1}(\{\text{false}\})$ . When  $\text{tr}$  is a layered valuation, if  $X \in \mathbf{T}_x$  or  $X \in \mathbf{F}_x$ , we say that  $x$  is true or false w.r.t.  $X$  respectively. By definition, the truth of  $x$  w.r.t.  $X$  depends only on the answers of lower level than  $\text{lev}(x)$ ; it follows that

$$\text{lev}(x) = 0 \Rightarrow \text{tr}(x, X) = \text{tr}(x, \emptyset)$$

that is, the truth value of answers of level 0 is absolute, and it depends just on the choice of  $\text{tr}$ . Level 0 answers play the role of “facts”. On the contrary answers of level greater than 0 are better seen as empirical hypotheses, that are considered true as far as they are not falsified by answers of lower level.

**Example 2** Continuing example 1, let us set  $\text{lev}(i, n, m) = i$ . Then we define the meaning of the answers in  $\mathbb{A}_2$  via the mapping  $\text{tr}$  by putting:  $\text{tr}((0, n, m), X) = \text{true}$  if and only if  $f_1(n) > f_1(m)$ , and:  $\text{tr}((1, n, m), X) = \text{true}$  if and only if  $n, m \in (\mathbb{N}_{f_1})^X$  and  $f_2(n) > f_2(m)$ . Recall that  $(\mathbb{N}_{f_1})^X$  is the set of  $n$  such that  $(0, n, p) \notin X$  for all  $p$ , and that if  $(i, n, m) \in \mathbb{A}_2 = \mathbb{A}_0 \uplus \mathbb{A}_1$  then  $n < m$ .

Clearly  $\text{tr}$  is layered since  $\text{tr}((i, n, m), X)$  does not depend on  $X$  when  $i = 0$ , while when  $i = 1$  it depends on  $X \upharpoonright 1 = \{(i, n, m) \in X \mid i = 0\}$ , which is morally  $\mathbb{A}_0 \cap X$ .

To see that  $\text{tr}$  is continuous let us observe that  $\text{tr}((0, n, m), X)$  is the constant function w.r.t.  $X$ , and that  $\text{tr}((1, n, m), X) = \text{true}$  if and only if  $f_2(n) > f_2(m)$  and  $n, m \in (\mathbb{N}_{f_1})^X$ , and the set of  $X$  for which this is true is a clopen as a consequence of Lemma 2.3.

From now on, we assume that some layered knowledge structure  $(\mathbb{A}, \sim, \text{lev}, \text{tr})$  has been fixed, with some level map  $\text{lev}$  and some continuous layered valuation  $\text{tr}$ . We now introduce the set  $\mathbb{S}$  of *sound* knowledge states (those from which nothing should be removed), the set  $\mathbb{C}$  of *complete* knowledge states (those to which nothing should be added), the set  $\mathbb{M}$  of *model* states (the “perfect” states, those from which nothing should be removed and to which nothing should be added). The present notion of “model” is different and much weaker than the Tarskian’s one; if atoms are to be logical statements, a “model” is a sound and complete theory; in general it defines the ideal subjective condition of having a full knowledge of the “world”, namely of a model in the familiar sense of logic.

**Definition 3.2** (Sound and Complete States). *Let  $X \in \mathbb{S}$ ,  $x \in \mathbb{A}$ . Then:*

- (1)  $X$  is sound if  $\forall x \in \mathbb{A}. x \in X \Rightarrow \text{tr}(x, X) = \text{true}$ ;
- (2)  $X$  is complete if  $\forall x \in \mathbb{A}. X \cap [x] = \emptyset \Rightarrow \text{tr}(x, X) = \text{false}$ ;
- (3)  $X$  is a model if it is sound and complete.

*We call  $\mathbb{S}$ ,  $\mathbb{C}$  and  $\mathbb{M}$  the sets of sound, complete and model states respectively.*

In words a state of knowledge  $X$  is sound if all the answers it contains are true w.r.t.  $X$  itself;  $X$  is complete if no answer which is true w.r.t.  $X$  and compatible with the answers in  $X$  can be consistently added to  $X$ ; hence  $X$  is a model if it is made of answers true w.r.t.  $X$  and it is maximal. We think of a model  $X$  as a perfect representation of the world. For

instance with respect to the examples in §1 and §2, if  $X$  is a model then the sets  $(\mathbb{N}_{f_1})^X$  and  $(\mathbb{N}_{f_1})_{f_2}^X$  are equal to  $(\mathbb{N}_{f_1})$  and  $(\mathbb{N}_{f_1})_{f_2}$  respectively, that is the beliefs of the agent perfectly agree with absolute truth.

In spite of this interpretation, models are far from being unique even w.r.t. a fixed map  $\text{tr}$ . Two models can include two different answers to the same question, because a question can have many true answers, while w.r.t. any state of knowledge each question is associated to a memory cell having room for a single answer.

Let us define  $\mathbf{S}_x = \{X \in \mathbb{S} \mid x \in X \Rightarrow \text{tr}(x, X) = \text{true}\}$ , or equivalently  $\mathbf{S}_x = (\mathbb{S} \setminus A_x) \cup \mathbf{T}_x$ ;  $\mathbf{C}_x = \{X \in \mathbb{S} \mid X \cap [x] = \emptyset \Rightarrow \text{tr}(x, X) = \text{false}\}$ , that is  $\mathbf{C}_x = (\mathbb{S} \setminus B_x) \cup \mathbf{F}_x$ , and  $\mathbf{M}_x = \mathbf{S}_x \cap \mathbf{C}_x$ . Clearly we have  $\mathbf{S} = \bigcap_{x \in \mathbb{A}} \mathbf{S}_x$ ,  $\mathbf{C} = \bigcap_{x \in \mathbb{A}} \mathbf{C}_x$  and  $\mathbf{M} = \bigcap_{x \in \mathbb{A}} \mathbf{M}_x$ .

From a topological viewpoint, it is interesting to observe that all the above subsets of  $\mathbb{S}$  are closed in  $\Omega(\mathbb{S})$ , while some of them are clopen.

**Lemma 3.3.** *For all  $x \in \mathbb{A}$ ,  $\mathbf{T}_x, \mathbf{F}_x, \mathbf{S}_x, \mathbf{C}_x, \mathbf{M}_x$  are clopen in  $\Omega(\mathbb{S})$ .  $\mathbf{S}, \mathbf{C}, \mathbf{M}$  are closed in  $\Omega(\mathbb{S})$ .*

*Proof.* For all  $x \in \mathbb{A}$ , the function  $\text{tr}(x) : \mathbb{S} \rightarrow 2$ , where  $\text{tr}(x)(X) = \text{tr}(x, X)$ , is continuous. Then by 2.3.3, and the fact that 2 is discrete, we deduce that  $\mathbf{T}_x, \mathbf{F}_x$  are clopen. By 2.3.1 and 2.3.2 the sets  $B_x, A_x$  are clopen, therefore  $\mathbf{S}_x, \mathbf{C}_x, \mathbf{M}_x$  are clopen as well, being obtained by complement, finite unions and intersections of clopen sets. We conclude that  $\mathbf{S}, \mathbf{C}, \mathbf{M}$  are closed because they are intersections of closed sets.  $\square$

It is immediate that sound sets exist, as well as complete ones: trivial examples are  $\emptyset$  which is vacuously sound, and any set  $X$  including one answer  $x$  for each equivalence class  $[x] \in \mathbb{Q}$ , which is vacuously complete, but not necessarily sound. Here is a non-trivial though simple example of these concepts.

**Example 3** Suppose that  $f_1(0) = 2 = f_1(n)$  for all  $n > 2$ , and that  $f_1(1) = 1$  and  $f_1(2) = 0$ . If we consider the states over  $\mathbb{A}_0$  only, and the restriction to  $\mathbb{A}_0$  of the mapping  $\text{tr}$  in Example 2, we have the models  $\{(0, 0, 1), (0, 1, 2)\}$  and  $\{(0, 0, 2), (0, 1, 2)\}$ . Any subset of these sets is sound, while  $\{(0, n, n+1) \mid n \in \mathbb{N}\}$  is complete but not sound.

It is not obvious, however, that models exist in general. The proof that models exist is somehow reminiscent of Gödel's proof that all first order consistent theories have a model.

**Theorem 3.4** (Existence of Models). *For every layered knowledge structure  $(\mathbb{A}, \sim, \text{lev}, \text{tr})$  whose space of knowledge is  $\mathbb{S}$ , there exists a model  $X \in \mathbb{S}$ .*

*Proof.* Fix an arbitrary indexing  $x_0, x_1, \dots$  of the countable set  $\mathbb{A}$ . For each  $x \in \mathbb{A}$  and  $Y \in \mathbb{S}$  set:

$$g(x, Y) = \begin{cases} \{x_i\} & \text{if } i \text{ is the minimum index } j \text{ s.t.} \\ & x_j \in [x] \wedge \text{tr}(x_j, Y) = \text{true}, \text{ if it exists} \\ \emptyset & \text{otherwise} \end{cases}$$

Now define inductively for each  $\gamma \in \text{Ord}$ :

$$X_\gamma = \bigcup \{g(x, X_{<\gamma}) \mid \text{lev}(x) = \gamma\} \text{ where } X_{<\gamma} = \bigcup_{\delta < \gamma} X_\delta$$

In words,  $X_\gamma$  is obtained by choosing an answer  $x'$ , if any, for each equivalence class  $[x]$  with  $\text{lev}(x) = \gamma$ , such that  $x'$  is true w.r.t. all the choices made at previous stages  $\delta < \gamma$ . Since  $x_i \in [x]$  implies that  $\text{lev}(x_i) = \text{lev}(x)$ ,  $X_\gamma$  is made of answers of level  $\gamma$ .

Then we prove that  $X = \bigcup_{\gamma \in \text{Ord}} X_\gamma$  is a model. First by construction  $X_\gamma$  is consistent for all  $\gamma$ , because it contains at most one answer for each equivalence class; this implies that  $X$  is consistent, since two answers in the same equivalence class are in the same  $X_\alpha$ . Second, if  $x \in X \upharpoonright \gamma$  then  $x \in X_{<\gamma}$ , so that:

$$\text{tr}(x, X) = \text{tr}(x, X \upharpoonright \text{lev}(x)) = \text{tr}(x, X_{<\text{lev}(x)}) = \text{true}$$

and  $X$  is sound. Finally, for all  $x \in \mathbb{A}$ , if  $X \cap [x] = \emptyset$  then

$$\begin{aligned} X \cap [x] = \emptyset &\Rightarrow X_{\text{lev}(x)} \cap [x] = \emptyset \\ &\Rightarrow \forall x' \in [x]. \text{tr}(x', X_{<\text{lev}(x)}) = \text{false} \\ &\Rightarrow \text{tr}(x, X_{<\text{lev}(x)}) = \text{false} \\ &\Rightarrow \text{tr}(x, X) = \text{false} \end{aligned}$$

by  $\text{tr}(x, X) = \text{tr}(x, X_{<\text{lev}(x)})$ . Therefore  $X$  is complete and hence a model.  $\square$

The construction of Theorem 3.4 is not effective, even when the layered knowledge structure is recursive. Assume that  $\gamma \in \text{Ord}$  is the number of levels of the knowledge structure. If we look closely to the proof, we see that we defined a model by some  $\Delta_{1+\gamma}^0$ -predicate. In particular, if there are infinitely many levels, then the definition is not an arithmetical predicate. We claim that the recursive complexity in our result is optimal: for any  $\gamma$  there is some recursive layered knowledge structure with  $\gamma$  levels, whose models are all (the extensions of)  $\Delta_{1+\gamma}^0$ -complete predicates, and therefore are never  $\Delta_{1+\delta}^0$ -predicates, for any  $\delta < \gamma$ . In general models are not recursive sets, and *a fortiori* not finite.

#### 4. INTERACTIVE REALIZABILITY

Given a layered knowledge structure  $(\mathbb{A}, \sim, \text{lev}, \text{tr})$ , the goal of a learning process is to reach some sound  $X \in \mathbb{S}$  which is sufficiently large to compute a solution to the problem at hand, e.g. a  $k$ -tuple  $n_1, \dots, n_k$  of natural numbers witnessing the truth of (1.1) or of (1.2) in §1. To make this precise, we formally define what it means that a *problem*  $P \subseteq C$ , (where  $C$  is an arbitrary denumerable set, the carrier) has a solution  $\alpha$  relative to a state  $X$ . Informally, we require that  $\alpha(X)$  is an element of  $C$  continuously depending on a knowledge state  $X$ , which belongs to  $P$  (or it satisfies the predicate  $P$ ) whenever  $X$  is a model. In the terminology of [4]  $\alpha$  is an “individual”.

We give definitions and results in full generality; however we are mainly interested in recursive  $\alpha$  and  $P$  and in effective ways of finding a state  $X$  such that  $\alpha(X) \in P$ .

**Definition 4.1** (Solution of a Problem w.r.t. a Knowledge Structure). *Let  $(\mathbb{A}, \sim, \text{lev}, \text{tr})$  be a layered knowledge structure and  $\mathbb{S}$  its space of states of knowledge. Given a continuous  $\alpha : \mathbb{S} \rightarrow C$  where  $C$  is taken with the discrete topology a predicate  $P \subseteq C$  (a problem), and  $X \in \mathbb{S}$ , we define:*

- (1)  $X \models_{\mathbb{A}} \alpha : P \Leftrightarrow \alpha(X) \in P$ ,
- (2)  $\models_{\mathbb{A}} \alpha : P \Leftrightarrow \forall X \in \mathbb{S}. X \text{ is a model} \Rightarrow X \models_{\mathbb{A}} \alpha : P$ .

When  $\models_{\mathbb{A}} \alpha : P$  we say that  $\alpha$  is a solution of  $P$  w.r.t.  $(\mathbb{A}, \sim)$ .

When  $P$  is a predicate defined by a formula, the set  $C$  is the carrier of a model in the usual sense; therefore  $\alpha$  is a multivalued individual akin to individuals in a product model. We shall omit the subscript  $\mathbb{A}$  in  $\models_{\mathbb{A}}$  when  $\mathbb{A}$  is understood; in the examples we take  $C = \mathbb{N}$ .

**Example 4** Let  $(\mathbb{A}_2, \sim_2)$  be the knowledge structure defined in example 1 in §2, and  $\mathbb{S}_2$  its knowledge space. Fix  $k \in \mathbb{N}$ ; writing  $\langle n_1, \dots, n_k \rangle$  for the code number of the  $k$ -tuple

$n_1, \dots, n_k$  we define the “problem”  $P_2$ :

$$P_2 = \{ \langle n_1, \dots, n_k \rangle \mid \bigwedge_{i < k} (n_i < n_{i+1} \wedge f_1(n_i) \leq f_1(n_{i+1}) \wedge f_2(n_i) \leq f_2(n_{i+1})) \},$$

$P_2$  is the set of all (coding of)  $k$ -tuple witnessing that (1.1) and (1.2) in §1 are true. Now for any  $X \in \mathbb{S}$  define:

$$\alpha_2(X) = \min \{ \langle n_1, \dots, n_k \rangle \mid n_1 < \dots < n_k \wedge n_1, \dots, n_k \in (\mathbb{N}_{f_1})_{f_2}^X \}$$

where  $\min$  is understood as the lexicographic ordering of the  $k$ -tuples. By definition the mapping  $\alpha_2$  picks the first  $k$  elements in the set  $(\mathbb{N}_{f_1})_{f_2}^X$  in increasing order. Note that  $\alpha_2$  is not a dummy search procedure, rather it is a primitive that reads of the guess for the solution to the problem  $P_2$  given the knowledge  $X$ .  $\alpha_2$  is always defined because  $\mathbb{N}_{f_1}^X$  and  $(\mathbb{N}_{f_1})_{f_2}^X$  are infinite for every  $X \in \mathbb{S}_2$ . Indeed this can be proved by a relativization to  $X$  of the argument of Lemma 1.1: if  $n \notin \mathbb{N}_{f_1}^X$  then there exists  $m \in \mathbb{N}$  s.t.  $n < m$  but  $f_1(n) > f_1(m)$  in the knowledge state  $X$ , namely we have  $(0, n, m) \in X$ . Were  $\mathbb{N}_{f_1}^X$  finite, we would be able to find infinitely many such  $m$  forming an infinite increasing chain, and so an infinite descending chain via  $f_1$ . Similarly one proves that  $(\mathbb{N}_{f_1})_{f_2}^X$  is infinite (quantifying over  $\mathbb{N}_{f_1}^X$  in place of  $\mathbb{N}$  and coding the counterexamples known at  $X$  by  $(1, n, m) \in X$ ).

We show that  $\alpha_2$  is continuous. Let  $\alpha_2(X) = \langle n_1, \dots, n_k \rangle$ : then  $n_i \in (\mathbb{N}_{f_1})_{f_2}^X$  for all  $i \leq k$ , and for all  $m < n_k$  with  $m \neq n_1, \dots, n_k$ , we have  $m \notin (\mathbb{N}_{f_1})_{f_2}^X$ . Conversely one can check that for all  $Y \in \mathbb{S}$ , if  $\bigwedge_{i \leq k} n_i \in (\mathbb{N}_{f_1})_{f_2}^Y$  and  $\bigwedge_{m < n_k, m \neq n_1, \dots, n_k} m \notin (\mathbb{N}_{f_1})_{f_2}^Y$  then  $\alpha_2(Y) = \langle n_1, \dots, n_k \rangle$ . But since we know that the predicate  $n \in (\mathbb{N}_{f_1})_{f_2}^Y$  is continuous in  $Y$  (see the remark after Lemma 2.3), the last condition defines a finite intersection of clopens, which is clopen.

We show that  $\alpha_2$  is a solution of  $P_2$ , that is, that  $\models \alpha_2 : P_2$ . Let  $X$  be a model: then we have  $(\mathbb{N}_{f_1})_{f_2}^X = (\mathbb{N}_{f_1})_{f_2}$ . Since  $\alpha(X) = \langle n_1, \dots, n_k \rangle \in (\mathbb{N}_{f_1})_{f_2}^X$ , we deduce  $\alpha_2(X) \in (\mathbb{N}_{f_1})_{f_2}$ , that is, that  $f_1$  and  $f_2$  are weakly increasing w.r.t.  $n_1, \dots, n_k$ . Thus,  $\alpha_2(X) \in P_2$ .

A solution  $\alpha$  is some way to produce an inhabitant  $\alpha(X) \in P$  out of any model  $X$ . A learning strategy for a problem  $P$  admitting a solution  $\alpha$  w.r.t.  $(\mathbb{A}, \sim, \text{lev}, \text{tr})$  is ideally a search procedure of some model  $X \in \mathbb{S}$ . But models are in general infinite and non-recursive states of knowledge: to make learning effective we rely on the continuity of  $\alpha$  which implies that if  $\alpha(X) = n \in P$  for some model  $X$  there exists a finite  $s \subseteq X$  such that  $\alpha(s) = n$ .

We describe the search of such finite sound approximations of a model  $X$  via certain continuous functions  $r$  over  $\mathbb{S}$ . The function  $r$  is such that for any sound  $X$  (not necessarily a model) the set  $r(X)$  is a finite set of answers that are not in  $X$  but are compatible with the answers in  $X$  and true w.r.t.  $X$ . So  $r$  is a mean to extend a state of knowledge with new answers; when  $r(X) = \emptyset$  there is nothing else to know and we expect that  $\alpha(X) \in P$ . When such a function exists for given  $P$  and  $\alpha$  we say that it is a *realizer* of the statement that  $\alpha$  is a solution of  $P$ . The name “realizer” comes from [6, 1, 4]; in particular in [1] it is shown how Kleene’s realizability can be extended to HA plus a restricted form of excluded middle by considering as realizers functionals depending on “states”, that are states of knowledge with just one level. Indeed the present concept subsumes that of realizers considered in the previous works.

**Definition 4.2** (Realizers and Zeros). *A realizer is a continuous map  $r : \mathbb{S} \rightarrow \mathcal{P}_{fin}(\mathbb{A})$  such that for all  $X \in \mathbb{S}$  and all  $x \in r(X)$ :*

- (1)  $X \cap [x] = \emptyset$ ,
- (2)  $\text{tr}(x, X) = \text{true}$ .

We denote by  $\mathcal{R}$  the set of realizers. Finally we say that  $X \in \mathbb{S}$  is a zero of  $r \in \mathcal{R}$  if  $r(X) = \emptyset$ .

For the sake of generality we do not assume any condition about the  $X$  above: in particular it is not necessarily sound.

We can see a realizer  $r$  as the essential part of a learning strategy, which tries to update the current state of knowledge. This is obtained by evaluating  $r(X)$  to get a finite set of new answers by which  $X$  could be soundly extended. To see this let  $\text{new} : \mathbb{S} \times \mathcal{P}_{fin}(\mathbb{A}) \rightarrow \mathcal{P}_{fin}(\mathbb{A})$  be defined by:

$$\text{new}(X, U) = \{x \in U \mid X \cap [x] = \emptyset \wedge \text{tr}(x, X) = \text{true}\}.$$

In words  $\text{new}(X, U)$  selects from  $U$  those atoms that can be consistently added to the state of knowledge  $X$ .

**Lemma 4.3.** *The function  $\text{new}$  is continuous, where  $\mathbb{S}$ ,  $\mathcal{P}_{fin}(\mathbb{A})$  and  $\mathbb{S} \times \mathcal{P}_{fin}(\mathbb{A})$  are taken with  $\Omega(\mathbb{S})$ , the discrete and the product topology respectively.*

*Proof.* Since  $\mathcal{P}_{fin}(\mathbb{A})$  has the discrete topology, it suffices to show that the set  $\{X \in \mathbb{S} \mid \text{new}(X, U) = V\}$  is open in  $\Omega(\mathbb{S})$  for all  $U, V \in \mathcal{P}_{fin}(\mathbb{A})$ . By definition of  $\text{new}$  for all  $x \in U$ , all  $Y \in \mathbb{S}$  we have  $\text{new}(Y, \{x\}) = \{x\}$  if and only if  $Y \in \mathbb{S} \setminus \mathbb{C}_x$ , and  $\text{new}(Y, \{x\}) = \emptyset$  if and only if  $Y \in \mathbb{C}_x$ . Let  $\mathcal{O} = (\bigcap_{x \in V} (\mathbb{S} \setminus \mathbb{C}_x)) \cap (\bigcap_{x \in (U \setminus V)} \mathbb{C}_x)$ . Then  $\mathcal{O}$  is a clopen, because it is obtained by complement and finite intersections of the sets  $\mathbb{C}_x$ , which are clopen by Lemma 3.3. By construction,  $Y \in \mathcal{O}$  if and only if  $\text{new}(Y, U) = V$ .  $\square$

Say that an *operator* over  $\mathbb{S}$  is any continuous map  $r : \mathbb{S} \rightarrow \mathcal{P}_{fin}(\mathbb{A})$ , and call  $Op_{\mathbb{S}}$ , or simply  $Op$ , the set of operators over  $\mathbb{S}$ . For  $r \in Op$  we set:  $\hat{r}(X) = \text{new}(X, r(X))$ .

**Lemma 4.4.** *The set of realizers is the subset of operators such that  $r = \hat{r}$ .*

**Proof.** By Lemma 4.3 and the fact that  $\hat{r} = \text{new} \circ \langle \text{id}_{\mathbb{S}}, r \rangle$ ,  $\hat{r}$  is continuous as soon as  $r$  is such. On the other hand by the very definition:

$$\hat{r}(X) = \text{new}(X, r(X)) = \{x \in r(X) \mid X \cap [x] = \emptyset \wedge \text{tr}(x, X) = \text{true}\}$$

Therefore  $\hat{r} \in \mathcal{R}$ . Beside if  $r \in \mathcal{R}$  then condition  $X \cap [x] = \emptyset \wedge \text{tr}(x, X) = \text{true}$  holds for all  $x \in r(X)$ , hence  $\hat{r} = r$ . From this and the fact that  $\hat{r} \in \mathcal{R}$  for all  $r \in Op$  it follows that  $\hat{\hat{r}} = \hat{r}$  for all operator  $r$ , and we are done.  $\square$

**Example 5** We propose a realizer solving the problem (1.1), expressed by the predicate:

$$P_1 = \{\langle n_1, \dots, n_k \rangle \mid \bigwedge_{i < k} (n_i < n_{i+1} \wedge f_1(n_i) \leq f_1(n_{i+1}))\}.$$

We first define a function  $\beta_1(X, \langle n_1, \dots, n_h \rangle)$  extending any list  $\langle n_1, \dots, n_h \rangle$  with  $h \leq k$  to a solution of (1.1):

$$\beta_1(X, \langle n_1, \dots, n_h \rangle) = \begin{cases} \langle n_1, \dots, n_h \rangle & \text{if } h = k \\ \beta_1(X, \langle n_1, \dots, n_h, m \rangle) & \text{where } m \text{ is the minimum s.t.} \\ & m \in \mathbb{N}_{f_1}^X \text{ and} \\ & m > n_h \text{ if } h > 0. \end{cases}$$

A function solving (1.1) may then be defined by  $\alpha_1(X) = \beta_1(X, \langle \rangle)$ . We claim that  $\models \alpha_1 : P_1$ . Indeed  $\beta_1$  is continuous w.r.t.  $X$  since  $m \in \mathbb{N}_{f_1}^X$  is equivalent to  $X \notin B_{(0, m, m+1)}$  which is a clopen by 2.3.1, so that  $\alpha_1$  is continuous. Further, if  $X$  is a model, then  $\alpha_1(X) \in P_1$ . We define a realizer  $r_1$  looking for some  $X$  such that  $\alpha_1(X)$  solves  $P_1$ .  $r_1(X)$  takes any knowledge state  $X$ , and adds to it the first counterexample to (1.1) we may find in the list  $\langle n_1, \dots, n_k \rangle$  generated by  $\alpha_1$ , unless  $\alpha_1(X)$  solves  $P_1$ . We define  $r_1$  in two steps: first, we define a map  $g_1$  finding the first counterexample to (1.1) in a given list:

$$g_1(\langle n_1, \dots, n_k \rangle) = \begin{cases} \{(0, n_i, n_{i+1}) \mid 1 \leq i < k \text{ min. s.t.} \\ \quad f_1(n_i) > f_1(n_{i+1})\} & \text{if } i \text{ exists} \\ \emptyset & \text{otherwise.} \end{cases}$$

Then we define the realizer by composing  $g_1$  with the output of  $\alpha_1$ :  $r_1(X) = g_1(\alpha_1(X))$ .  $r_1$  is a realizer, because  $r_1(X)$  always outputs atoms not in  $X$ . Indeed, if  $\alpha_1(X) = \langle n_1, \dots, n_k \rangle \in \mathbb{N}_{f_1}^X$ , and if  $r_1(X)$  output the atom  $f_1(n_i) > f_1(n_{i+1})$ , then  $(f_1(n_i) > f_1(n_{i+1})) \notin X$  by definition of  $\langle n_1, \dots, n_k \rangle \in \mathbb{N}_{f_1}^X$ . We may use  $r_1(X)$  to extend  $X$  until we find some  $X$  such that  $r_1(X) = \emptyset$ . Whenever  $r_1(X) = \emptyset$  we have  $g_1(\alpha_1(X)) = \emptyset$ , hence  $\alpha_1(X)$  solves (1.1) by definition of  $g_1$ .

To step from  $P_1$  to  $P_2$ , namely to problem (1.2), we just replace  $\mathbb{N}_{f_1}^X$  by  $\mathbb{N}_{f_1, f_2}^X$ , namely:

$$\beta_2(X, \langle n_1, \dots, n_h \rangle, k) = \begin{cases} \langle n_1, \dots, n_h \rangle & \text{if } h = k \\ \beta_2(X, \langle n_1, \dots, n_h, m \rangle, k) & \text{where } m \text{ is the minimum s.t.} \\ & m \notin (\mathbb{N}_{f_1})_{f_2}^X \\ & \text{and } m > n_h \text{ if } h > 0. \end{cases}$$

and

$$g_2(\langle n_1, \dots, n_k \rangle) = \begin{cases} \{(0, n_i, n_{i+1}) \mid 1 \leq i < k \text{ min. s.t.} \\ \quad f_1(n_i) > f_1(n_{i+1}) \vee f_2(n_i) > f_2(n_{i+1})\} & \text{if } i \text{ exists} \\ \emptyset & \text{otherwise.} \end{cases}$$

We have that  $\alpha_2(X) = \beta_2(X, \langle \rangle, k)$ , where  $\alpha_2$  is from example 4. Now let us define  $r_2(X) = g_2(\beta_2(X, \langle \rangle, k)) = g_2(\alpha_2(X))$ . Then we can show that  $r_2$  is a realizer looking for some  $X$  such that  $\alpha_2(X) \in P_2$  just as in the case of  $r_1$  above.

If  $X$  is a model and  $r \in \mathcal{R}$  then  $r(X) = \emptyset$  by definition; on the other hand if  $\models_{\mathbb{A}} \alpha : P$  then  $\alpha(s) = \alpha(X) = n \in P$  for some finite  $s \subseteq X$ . Since  $r$  is continuous, the condition that reveals that the approximation  $s$  of  $X$  is good enough to compute an  $n \in P$  is that  $r(s) = r(X) = \emptyset$ . This suggests that a constructive way to meet the requirement about models in the definition of  $\models_{\mathbb{A}} \alpha : P$  is to ask that sound zeros of a realizer  $r$  are enough to find inhabitants of  $P$  via  $\alpha$ , and then look for finite sound zeros of  $r$ . We turn this into the following definition.

**Definition 4.5** (Interactive Realizability). *Let  $\alpha : \mathbb{S} \rightarrow C$  be continuous (w.r.t. the discrete topology over  $\mathbb{N}$ ), and  $P \subseteq C$  a predicate over the carrier  $C$ :*

(1)  *$r \in \mathcal{R}$  interactively realizes  $P$  w.r.t.  $\alpha$ , written  $r \vdash \alpha : P$ , if and only if:*

$$\forall X \in \mathbb{S}. X \text{ sound zero of } r \Rightarrow \alpha(X) \in P$$

(2)  *$P$  is interactively realizable w.r.t.  $\alpha$ , written  $\vdash \alpha : P$ , if and only if:*

$$\exists r \in \mathcal{R}. r \vdash \alpha : P.$$

If  $P_1$  and  $P_2$  are the predicates with carrier  $\mathbb{N}$  defined in examples 4 and 5,  $\alpha_1, \alpha_2$  their respective solutions and  $r_1, r_2$  the realizers from example 5; then it can be proved that  $r_i \vdash \alpha_i : P_i$  for both  $i = 1, 2$ .

The main result of the paper is that the apparently stronger  $r \vdash \alpha : P$  for some  $r \in \mathcal{R}$  is equivalent to  $\models \alpha : P$ . That is, whenever  $\alpha : P$  is valid then it is interactively learnable, and we have some strategy to find some finite  $X$  such that  $\alpha(X) \in P$ .

Before we establish the existence of sound finite zeros of any  $r \in \mathcal{R}$ . This is a non trivial fact because, whenever we add to some state  $Y$  (no matter whether finite or not) a  $y \in r(Y)$ , we know that  $\text{tr}(z, Y) = \text{true}$  for all  $z \in Y$ , but we do not know about the value of  $\text{tr}(z, Y \cup \{y\})$ , so that  $Y \cup \{y\}$  is not necessarily sound. Moreover it is not true that if  $s \subseteq X$  and  $X$  is sound then  $s$  is sound.

**Example 6** Let us redefine  $f_1$  by  $f_1(0) = 10, f_1(1) = 30, f_1(2) = 20$  and define  $f_2(0) = 20, f_2(1) = 10, f_2(2) = 20$ . Also we let  $x = (1, 0, 1)$  meaning that  $0 \notin (\mathbb{N}_{f_1})_{f_2}$ , and  $y = (0, 1, 2)$  meaning that  $1 \notin \mathbb{N}_{f_1}$ . Then  $\text{tr}(x, \{x\}) = \text{true}$  since at  $\{x\}$  it is likely that  $0 \notin (\mathbb{N}_{f_1})_{f_2}$  because of the counterexample in the point 1; but  $\text{tr}(x, \{x, y\}) = \text{false}$  because the discovery that  $1 \in \mathbb{N}_{f_1}$  contradicts counterexample on point 1.

We prove below that finite sound zeros exist for all  $r \in \mathcal{R}$  and that these are finite approximations of sound states of knowledge which are themselves zeros of  $r$ , hence in particular of models.

**Lemma 4.6.** *If  $X \in \mathbb{S}$  is sound,  $s \in \mathbb{S}_{fn}$  is a finite state such that  $s \subseteq X$ , then there exists a finite sound  $t \in \mathbb{S}_{fn}$  such that  $s \subseteq t \subseteq X$ .*

*Proof.* First let us extend the definition of  $\text{lev}$  to (finite) states:  $\text{lev}(s) = \max\{\text{lev}(x) \mid x \in s\}$  if  $s \neq \emptyset$ ,  $\text{lev}(\emptyset) = 0$ . Now by induction over  $\text{lev}(s)$ , we prove that if  $s \subseteq X$ , then there exists  $t \subseteq X$  which is finite, sound and such that  $s \subseteq t$ ; moreover  $t$  can be chosen such that  $\text{lev}(t) = \text{lev}(s)$ .

If  $\text{lev}(s) = 0$  then for all  $x \in s$  it is the case that

$$\text{tr}(x, s) = \text{tr}(x, \emptyset) = \text{tr}(x, X) = \text{true}$$

because of  $\text{lev}(x) = 0$ . Hence in this case  $t = s$ .

Let  $\text{lev}(s) = \alpha > 0$ : then let  $s' = \{x \in s \mid \text{lev}(x) = \alpha\}$  and  $s'' = s \setminus s'$ . If  $s'' = \emptyset$  then  $\text{lev}(s'') = 0 < \alpha$  by definition; otherwise  $\text{lev}(s'') = \max\{\text{lev}(y) \mid y \in s''\} < \alpha$  by construction.

Suppose that  $s' = \{x_1, \dots, x_k\}$ : since  $s' \subseteq s \subseteq X$  we know that

$$\text{tr}(x_i, X) = \text{tr}(x_i, X \upharpoonright \alpha) = \text{true}$$

for all  $1 \leq i \leq k$ . By continuity, for all  $i$  there exists a basic open  $\mathcal{O}_{s_i, t_i}$  s.t.  $X \upharpoonright \alpha \in \mathcal{O}_{s_i, t_i}$  and  $\text{tr}(x_i, \mathcal{O}_{s_i, t_i}) = \text{true}$ . It follows that  $\mathcal{O} = \mathcal{O}_{s_1, t_1} \cap \dots \cap \mathcal{O}_{s_k, t_k}$  is open,  $X \upharpoonright \alpha \in \mathcal{O}$  and  $\text{tr}(x_i, \mathcal{O}) = \text{true}$  for all  $i$ . Let  $U = s_1 \cup \dots \cup s_k \subseteq X \upharpoonright \alpha$  and  $V = t_1 \cup \dots \cup t_k$ ; since  $X \upharpoonright \alpha \in \mathcal{O}$ ,



we know that  $(X \upharpoonright \alpha) \cap V = \emptyset$ . Thus, for any  $t$  s.t.  $U \subseteq t \subseteq X \upharpoonright \alpha$  it has to be the case that  $t \in \mathcal{O}$ , and therefore  $\text{tr}(x_i, t) = \text{true}$  for all  $i$ .

Let  $W = U \cup s'' \subseteq X \upharpoonright \alpha$ , so that  $\text{lev}(W) = \beta < \alpha$ , because  $U \subseteq X \upharpoonright \alpha$ . By induction there exists a sound and finite  $t'$  such that  $t' \subseteq X$ ,  $\text{lev}(t') = \beta$  and  $W \subseteq t' \subseteq X \upharpoonright \alpha$ .

By construction  $s'' \subseteq t'$ , so that  $s \subseteq t' \cup s' \subseteq X$  and  $\text{lev}(t' \cup s') = \alpha$ . On the other hand  $t' \cup s'$  is sound, since for all  $y \in t' \cup s'$ , either  $y \in t'$ , and then

$$\text{tr}(y, t' \cup s') = \text{tr}(y, t') = \text{true}$$

because  $\text{lev}(s') = \alpha > \text{lev}(t') \geq \text{lev}(y)$  and  $t'$  is sound. Otherwise, if  $y \in s'$  we have that  $\text{lev}(y) = \alpha = \text{lev}(s')$  so that

$$\text{tr}(y, t' \cup s') = \text{tr}(y, (t' \cup s') \upharpoonright \alpha) = \text{tr}(y, t')$$

But since  $t' \in \mathcal{O}$  by the above, we conclude that  $\text{tr}(y, t') = \text{true}$ .  $\square$

We are now in place to conclude the proof that every realizer has a finite sound zero.

**Theorem 4.7** (Existence of Sound and Finite Zeros of Realizers). *If  $r \in \mathcal{R}$ , then there exists a finite sound zero  $t \in \mathbb{S}_{\text{fin}}$  of  $r$ .*

*Proof.* Models exist by Theorem 3.4 and they are sound by definition, hence  $r(X) = \emptyset$  for some sound  $X \in \mathbb{S}$  since  $r \in \mathcal{R}$ . By continuity there is a basic open  $\mathcal{O}_{s_0, t_0}$  such that  $X \in \mathcal{O}_{s_0, t_0}$  and  $r(\mathcal{O}_{s_0, t_0}) = \emptyset$ . This implies that  $s_0 \subseteq X$  and  $X \cap t_0 = \emptyset$ , so that a fortiori any finite  $t \in \mathbb{S}_{\text{fin}}$  such that  $s_0 \subseteq t \subseteq X$  satisfies  $t \cap t_0 = \emptyset$  and therefore  $t \in \mathcal{O}_{s_0, t_0}$ , i.e. it is a zero of  $r$ . By Lemma 4.6 there exists a sound  $t$  among them, which is the desired finite sound zero of  $r$ .  $\square$

We come now to the completeness theorem. Our thesis is that interactive realizability is complete in the sense that if  $\alpha(X) \in P$  for all models  $X$ , then we may replace the model  $X$  by the finite sound zeros of a suitable realizer  $r \in \mathcal{R}$ . The proof of completeness is somehow reminiscent of Gödel's proof that valid formulas are provable, with realizers in the place of classical proofs.

**Theorem 4.8** (Completeness of Realization). *For any continuous  $\alpha : \mathbb{S} \rightarrow C$  and predicate  $P \subseteq C$  over the carrier  $C$ :*

- (1)  $\vdash \alpha : P$  if and only if  $\models \alpha : P$ .
- (2) If  $\models \alpha : P$  then the realizer  $r$  such that  $r \vdash \alpha : P$  can be chosen recursive relatively to  $\alpha$  and  $P$ .

*Proof.* We prove (1) and (2) simultaneously.  $(\Rightarrow)$  If  $X \in \mathbb{S}$  is a model then  $X$  is a sound zero of any realizer by Definition 4.2; hence if  $r \vdash \alpha : P$  for some  $r \in \mathcal{R}$  we immediately have  $\alpha(X) \in P$ , i.e.  $X \models \alpha : P$  for arbitrary model  $X$ .

$(\Leftarrow)$  We have to show that, if  $\models \alpha : P$ , namely if  $\alpha(X) \in P$  for  $X \in \mathbb{M}$ , then there exists an  $r \in \mathcal{R}$  recursive in  $\alpha$  and  $P$ , such that  $r \vdash \alpha : P$ . We establish the contrapositive:

$$\alpha(X) \notin P \Rightarrow X \text{ not sound} \vee r(X) \neq \emptyset$$

for some realizer  $r$  and arbitrary  $X \in \mathbb{S}$ .

If  $\alpha(X) \notin P$  then, by the hypothesis,  $X \notin \mathbb{M}$ , hence  $X \notin \mathbb{S}$  or  $X \notin \mathbb{C}$ . By definition of  $\mathbb{S}$  and  $\mathbb{C}$ , this implies that  $\exists x \in \mathbb{A}. X \notin \mathbb{S}_x \vee X \notin \mathbb{C}_x$ . Fix an enumeration  $x_0, x_1, \dots$  of the countable set  $\mathbb{A}$ . Let us define  $r : \mathbb{S} \rightarrow \mathcal{P}_{\text{fin}}(\mathbb{A})$  recursive in  $\alpha$  and  $P$ :

$$r(X) = \begin{cases} \emptyset & \text{if } \alpha(X) \in P \\ \{x_i\} & \text{where } i = \min\{j \in \mathbb{N} \mid X \in \mathbb{S} \setminus \mathbb{M}_{x_j}\}, \text{ else.} \end{cases}$$

Then  $r$  is a total function since if  $\alpha(X) \notin P$  then  $X \in \mathbb{S} - \mathbb{M}$  so that  $\{x_j \in \mathbb{A} \mid X \in \mathbb{S} \setminus \mathbb{M}_{x_j}\} \neq \emptyset$ . If  $r$  is continuous then  $\hat{r}(X) = \text{new}(X, r(X))$  because  $X \in \mathbb{S} \setminus \mathbb{M}_{x_j}$  implies  $X \in \mathbb{S} \setminus \mathbb{C}_{x_j}$ . Thus,  $r$  is a realizer by Prop. 4.4. We have  $\hat{r} \vdash \alpha : P$ . Indeed, assume for contradiction that  $\alpha(X) \notin P$ ,  $X \in \mathbb{S}$  and  $\hat{r}(X) = \emptyset$ . Then  $r(X) = \{x_i\}$  and  $X \in \mathbb{S} \setminus \mathbb{M}_{x_i} = (\mathbb{S} \setminus \mathbb{S}_{x_i}) \cup (\mathbb{S} \setminus \mathbb{C}_{x_i})$ . Since  $X \in \mathbb{S} \subseteq \mathbb{S}_{x_i}$ , then  $X \in \mathbb{S} \setminus \mathbb{C}_{x_i}$ . We conclude that  $\hat{r}(X) = \{x_i\}$ , contradiction.

To see that  $r$  is continuous it suffices to check that both  $r^{-1}(\emptyset)$  and  $r^{-1}(\{x\})$  (for any  $x \in \mathbb{A}$ ) are opens in  $\Omega(\mathbb{S})$ . Now  $r(X) = \emptyset$  if and only if  $\alpha(X) \in P$ , that is  $X \in \alpha^{-1}(P)$  which is clopen by Lemma 3.3. On the other hand  $X \in r^{-1}(\{x\})$  if and only if:

$$\exists i. x_i = x \wedge X \in (\mathbb{S} \setminus \mathbb{M}_{x_i}) \wedge \forall j < i. X \in \mathbb{M}_{x_j}.$$

This is equivalent to  $X \in \mathbb{M}_{x_0} \cap \dots \cap \mathbb{M}_{x_{i-1}} \cap (\mathbb{S} \setminus \mathbb{M}_{x_i})$  which, by Lemma 3.3, is a finite intersection of clopens, hence a clopen itself.  $\square$

## 5. AN ALGORITHM TO COMPUTE FINITE SOUND ZEROS OF REALIZERS

Theorem 4.7 establishes that any realizer has a finite sound zero, which can be found by blind search. In this section we address the question of a realistic computation of one zero given an effective realizer over an effective knowledge structure.

Let  $(\mathbb{A}, \sim, \text{lev}, \text{tr})$  be a layered knowledge structure, and assume for simplicity that the image of  $\mathbb{A}$  by  $\text{lev}$  is just the ordinal 2, namely that there are only two levels. Let  $\mathbb{S}$  and  $\mathbb{S}_{\text{fin}}$  denote the space of knowledge states and of finite knowledge states over  $\mathbb{A}$  respectively. Recall that given an answer  $x \in \mathbb{A}$  we say that it is a *fact* if  $\text{lev}(x) = 0$ , and a *hypothesis* if  $\text{lev}(x) = 1$ .

Suppose that  $\mathbb{A}$  and  $\sim$  are decidable, and  $\text{lev}$  and  $\text{tr}$  are computable. Suppose also that  $r \in \mathcal{R}$  is a realizer for this structure, and that it is computable. To compute a (finite) sound zero of  $r$  a tentative algorithm, generalizing what we have seen in §1, is as follows:

```

X := any finite sound state of knowledge, e.g.  $\emptyset$ 
while  $r(X) \neq \emptyset$  let  $x \in r(X)$ 
  Y :=  $\{x \in X \mid \text{tr}(x, X \cup \{x\}) = \text{true}\}$ 
  X :=  $Y \cup \{x\}$ 
return X

```

The idea is to compute the new answers  $r(X)$  that are compatible and true at  $X$ , and to choose non deterministically some  $x \in r(X)$ , whenever it exists, and to add  $x$  to  $X$  until we reach some  $X'$  such that nothing can be added to  $X'$  (that is such that  $r(X') = \emptyset$ ). The intermediate step of computing  $Y \subseteq X$  is needed because if we add  $x$  we might falsify some hypothesis  $y \in X$  (i.e., some  $y$  of level 1), and therefore we have to remove it. This is why the subsequent values of  $X$  do not form a chain w.r.t. subset inclusion in general, and we dub the whole process “non-monotonic learning”.

However the tentative algorithm does not always terminate: this is due to the fact that, before being removed,  $y$  could be used by  $r$  to add some new hypothesis  $y'$  to  $X$ , then  $y'$  is removed only after  $r$  uses  $y'$  to add some new hypothesis  $y''$  to  $X$ , and so forth. The only way to prevent divergence is to remove  $y$  together with all the hypothesis that have been added in any previous step *because of*  $y$ . For the sake of simplicity we add an integer variable to keep track of “time”, and remove from  $X$  all the  $z$  that have been added after  $y$ , which include all those logically depending on  $y$ , thought do not necessarily coincide with

them. By using simple sets of pairs  $(x, n)$  where  $x \in \mathbb{A}$  and  $n \in \mathbb{N}$  we record at what step an answer has been added to the current state of knowledge, and we use this information while removing hypothesis from the state  $X$ . The algorithm is modified as follows:

**Algorithm Find Sound Zero:**

1. input any finite sound state of knowledge  $X \in \mathbb{S}_{fin}$  and computable  $r \in \mathcal{R}$
2.  $D := \{(x, 0) \mid x \in X\}$  and  $n := 0$
3. while  $r(X) \neq \emptyset$  let  $x \in r(X)$
4.  $E := \{(y, m) \in A \mid \neg \exists (z, k) \in D. \text{lev}(z) \leq \text{lev}(y) \wedge k \leq m \wedge \text{tr}(z, X \cup \{x\}) = \text{false}\}$
5.  $D := E \cup \{(x, n)\}$  and  $n := n + 1$
6.  $X := \{z \in \mathbb{A} \mid \exists i \in \mathbb{N}. (z, i) \in D\}$
7. return  $X$

Let  $X_0, X_1, \dots$  be the subsequent values of the program variable  $X$ ; then we claim that the following is an invariant of the algorithm loop:

**Lemma 5.1.** *For all  $i$ ,  $X_i \in \mathbb{S}_{fin}$  and it is sound.*

*Proof.* The thesis holds for  $X_0$  by assumption. Assume that  $X_i$  is a finite sound state of knowledge and consider  $X_{i+1}$ . Because of the assignment in step 4. we know that  $E \subseteq D$ , so that by the assignments in steps 5. and 6.,  $X_{i+1} \subseteq X_i \cup \{x\}$ . By the fact that  $r$  is a realizer we know that  $X_i \cap [x] = \emptyset$ , and therefore  $X_i \cup \{x\} \in \mathbb{S}_{fin}$ , so that  $X_{i+1} \in \mathbb{S}_{fin}$ .

To see that  $X_{i+1}$  is sound, let us observe that at step 4. a pair  $(z, k)$  cannot be removed from  $D$  if  $\text{lev}(z) = 0$ , because  $\text{tr}(z, X_i) = \text{true}$  by the soundness of  $X_i$  and  $\text{tr}(z, X_i \cup \{x\}) = \text{tr}(z, (X_i \cup \{x\}) \upharpoonright 0) = \text{tr}(z, \emptyset) = \text{tr}(z, X_i) = \text{true}$ , because in that case  $z$  is a fact. Consider now the case of  $z \in X_{i+1}$  with  $\text{lev}(z) = 1$ , namely when  $z$  is a hypothesis. Since we know that  $X_{i+1} \subseteq X_i \cup \{x\}$ , we have that  $\text{tr}(z, X_i) = \text{true}$ , either because  $z \in X_i$  which is sound, or because  $z = x$  and  $r$  is a realizer. If  $\text{lev}(x) = 1$  then by the above remark that no atom of level 0 can be removed from  $X_i$ , we have that  $X_{i+1} \upharpoonright 1 = X_i \upharpoonright 1$  which implies that  $\text{tr}(z, X_{i+1}) = \text{tr}(z, X_{i+1} \upharpoonright 1) = \text{tr}(z, X_i \upharpoonright 1) = \text{tr}(z, X_i) = \text{true}$ . Therefore the only interesting case is when  $\text{lev}(z) = 1$  and  $\text{lev}(x) = 0$  where  $x$  is the atom newly chosen in  $r(X_i)$ , because this is right the case in which the valuation of a hypothesis might change. However we observe that in this case  $X_{i+1} \upharpoonright 1 = X_i \upharpoonright 1 \cup \{x\}$ , and that if  $z \in X_{i+1}$  then  $(z, m) \in E$  for some  $m$ ; it follows that  $\text{tr}(z, X_{i+1}) = \text{tr}(z, X_{i+1} \upharpoonright 1) = \text{tr}(z, X_i \upharpoonright 1 \cup \{x\}) = \text{tr}(z, X_i \cup \{x\}) \neq \text{false}$  by the condition in step 4. and hence  $\text{tr}(z, X_{i+1}) = \text{true}$ .  $\square$

It remains to show that the above algorithm always terminates. Before embarking into the proof let us recall that, in general, the sequence  $X_0, X_1, \dots$  is not increasing w.r.t. subset inclusion, because hypothesis appearing in the earlier stages can be removed at step 4. from subsequent ones. Nonetheless we have that  $X_0 \upharpoonright 1 \subseteq X_1 \upharpoonright 1 \subseteq \dots$ , so that by the continuity of the realizer  $r$  we can argue that the values of  $r$  over that chain are eventually constant:

**Lemma 5.2.** *If  $Y_0 \subseteq Y_1 \subseteq \dots$  is a non decreasing chain of states in  $\mathbb{S}$  and  $r \in \mathcal{R}$  then there is an  $n_0$  such that for all  $n \geq n_0$ ,  $r(Y_n) = r(Y_{n_0})$ .*

*Proof.* Let  $Y = \bigcup_n Y_n$ . Then  $Y \in \mathbb{S}$  since, by the hypothesis that  $Y_0 \subseteq Y_1 \subseteq \dots$ , any  $x, y \in Y$  belong to some  $Y_n$  which implies that either  $x = y$  or  $x \not\sim y$  (that is  $x \# y$ ). By continuity of  $r$  there is a basic open  $\mathcal{O}_{U,V} = \bigcap_{x \in U} A_x \cap \bigcap_{y \in V} B_y$ , with finite  $U, V$ , such that

$Y \in \mathcal{O}_{U,V}$  and  $r(\mathcal{O}_{U,V}) = r(Y)$ . The hypothesis that  $\{Y_i\}$  is a chain and that  $U$  is finite implies that there is  $n_0$  such that for all  $n \geq n_0$ ,  $U \subseteq Y_n$ ; on the other hand, for all  $z \in V$ , it is the case that  $Y \cap [z] = \emptyset$  so that  $Y_n \cap [z] = \emptyset$  since  $Y_n \subseteq Y$ . We conclude that for all  $n \geq n_0$  it is  $Y_n \in \mathcal{O}_{U,V}$  and therefore  $r(Y_n) = r(Y) = r(Y_{n_0})$ .  $\square$

As observed above, the  $X_i$  do not form a chain w.r.t. subset inclusion; however we can extract a subsequence of  $X_0, X_1, \dots$  such that the monotonicity condition does hold. Let  $D_i$  and  $E_i$  be the values of  $D$  and  $E$  after  $i$  steps of the algorithm respectively. In case  $r(X_i) = \emptyset$  we set  $X_j = X_i$ ,  $D_j = D_i$  and  $E_j = E_i$  for all  $j > i$ , so that the  $X_i$ ,  $D_i$  and  $E_i$  are always infinite sequences of (finite) sets, and if  $r(X_i) = \emptyset$  for some  $i$  then the sequences are definitely constant and the sets  $r(X_j)$  are definitely empty.

Let  $D_\infty = \{(x, k) \mid \forall i \geq k. (x, k) \in D_i\}$ , namely the set of pairs  $(x, k)$  which are definitely elements of the  $D_i$ . Also consider the statements:

$$(\text{Eq}_i) : \quad E_i = \{(x, k) \in D_\infty \mid k < i\}.$$

Then we prove in two steps that the equations  $(\text{Eq}_i)$  are true for infinitely many  $i$ .

**Lemma 5.3.** *For all  $i$  we have  $\{(x, k) \in D_\infty \mid k < i\} \subseteq E_i$ .*

*Proof.* If  $(x, k) \in D_\infty$  then  $(x, k) \in D_k$ ; if  $k < i$  but  $(x, k) \notin E_i$  then  $(x, k) \notin D_i$  since  $D_i = E_i \cup \{(z, i)\}$  by the assignment 5. and even if  $z = x$  it will be  $i \neq k$ . By this we get the contradiction  $(x, k) \notin D_\infty$ .  $\square$

**Lemma 5.4.** *For all  $i$  there is  $j \geq i$  such that  $(\text{Eq}_j)$  holds.*

*Proof.* Let  $\Delta_i = E_i \setminus \{(x, k) \in D_\infty \mid k < i\}$ , that is the set of the pairs in  $E_i$  that will be removed at stages later than  $i$ . By Lemma 5.3  $\{(x, k) \in D_\infty \mid k < i\} \subseteq E_i$ , hence if  $\Delta_i = \emptyset$  then  $(\text{Eq}_i)$  holds. Suppose instead that  $\Delta_i \neq \emptyset$  and consider  $(y, h) \in \Delta_i$  with minimal  $h$ .  $(y, h)$  is minimal in any  $\Delta_j$  with  $j \geq i$  to which it belongs, because the elements added to  $\Delta_i$  have the form  $(z, p)$  for some  $p > i$ .

This implies that  $h < i$  because  $(y, h) \in E_i$ , and there exists  $j > i$  such that  $(y, h) \notin D_j$  because  $(y, h) \notin D_\infty$ .

But if  $D_j$  is the first such set, we have that  $\text{tr}(z, X_j \cup \{x_j\}) = \text{false}$  (where  $x_j$  is the answer chosen at instruction 3. of step  $j$ ) for some  $(z, k) \in D_j$  such that  $\text{lev}(z) \leq \text{lev}(y)$  and  $k \leq h$ . If we remove  $(y, h)$ , for the minimality of  $h$  in  $\Delta_j$  and the condition in the instruction 4. we remove all elements of  $\Delta_j$ . Therefore  $\Delta_j = \emptyset$ , and hence  $(\text{Eq}_j)$  holds.  $\square$

Let  $X_{i_0}, X_{i_1}, \dots$  be the subsequence of  $X_0, X_1, \dots$  such that  $(\text{Eq}_{i_k})$  holds for all  $i_k$ . This is an infinite sequence by Lemma 5.4, and it is a non decreasing chain w.r.t. subset inclusion by construction.

**Lemma 5.5.**

- (1) *For some  $k_0, z_1, \dots, z_n$  and for all  $k \geq k_0$  we have  $r(X_{i_k}) = \{z_1, \dots, z_n\}$ .*
- (2) *For every  $x \in \mathbb{A}$  either  $\text{tr}(x, X_i) = \text{true}$  definitely or  $\text{tr}(x, X_i) = \text{false}$  definitely.*
- (3) *There are  $k_1, \dots, k_n \geq k_0$  such that  $\text{tr}(z_i, X_i) = \text{true}$  for all  $i \geq k_j$ .*
- (4) *For all  $k \geq k_j$  if  $z_j \in X_{i_{k+1}} \setminus X_{i_k}$  then  $(z_j, i_k) \in D_\infty$ .*
- (5)  *$r(X_{i_k}) = \emptyset$  for all  $k \geq \max(k_1, \dots, k_n)$ .*

*Proof.* Part (1) follows by Lemma 5.2 and the fact that the subsequence  $X_{i_0}, X_{i_1}, \dots$  is a non decreasing chain.

Part (2) is a consequence of the fact that  $\text{tr}(x, \_)$  is continuous in its second argument, and it is layered, so that it depends only on the answers in the chain  $X_0 \upharpoonright 1 \subseteq X_1 \upharpoonright 1 \subseteq \dots$ .

Part (3): by part (1) and Lemma 5.4, every  $z_j$  belongs to infinitely many  $r(X_i)$ , since  $X_{i_0}, X_{i_1}, \dots$  is a subsequence of  $X_0, X_1, \dots$ . Then, by (2), we have  $\text{tr}(z_j, X_i) = \text{true}$  definitively. Indeed we cannot have  $\text{tr}(z_j, X_i) = \text{false}$  definitively since  $r$  is a realizer, and therefore  $\text{tr}(z_j, X_{i_k}) = \text{true}$  for all  $k$ .

Part (4): assume  $k \geq k_j$  and  $z_j \in X_{i_k+1} \setminus X_{i_k}$ , in order to prove  $(z_j, i_k) \in D_\infty$ . To this aim, we assume  $i \geq i_k + 1$  in order to prove  $(z_j, i_k) \in D_i$ . By definition of  $D_i$ , we have to prove that if  $D_i \setminus D_{i-1} = (x_{i-1}, i-1)$ , then for all  $(y, h) \in D_i$  with  $h \leq i_k$  we have  $\text{tr}(y, X_{i-1} \cup \{x_i\}) = \text{true}$ . If  $h < i_k$ , then, since  $i \leq i_k + 1$ , any  $(y, h) \in D_i$  is in  $X_{i_k} \subseteq D_\infty$ , therefore  $\text{tr}(y, X_{i-1} \cup \{x_i\}) = \text{true}$ . Assume  $h = i_k$ , hence  $y = z_j$ : we have to prove that  $\text{tr}(z_j, X_{i-1} \cup x_i) = \text{true}$ . From the fact that  $\text{tr}$  is stratified we deduce  $\text{tr}(z_j, X_{i-1} \cup x_i) = \text{tr}(z_j, X_{i-1} \cup x_i \upharpoonright 1) = \text{tr}(z_j, X_i \upharpoonright 1) = \text{tr}(z_j, X_i) = \text{true}$  by the choice of  $z_j$ .

Part (5): let  $r(X_{i_k}) = \{z_1, \dots, z_n\}$  be the definitely constant value of  $r$  over the chain  $X_{i_0}, X_{i_1}, \dots$ , existing by part (1). By contradiction suppose that  $n > 0$ : by the choice of instruction 3. of the algorithm there is some  $z_j \in X_{i_k+1} \setminus X_{i_k}$ . By Part (4) we have  $(z_j, i_k) \in D_\infty$ , therefore  $z_j \in X_{i_h} \cap r(X_{i_h})$  for all  $h$  large enough, contradicting the fact that  $r$  is a realizer.  $\square$

Lemma 5.5 (5) and the fact that  $X_{i_0}, X_{i_1}, \dots$  is an infinite subsequence of  $X_0, X_1, \dots$  implies the termination of the algorithm. In summary we have:

**Theorem 5.6.** *If  $r \in \mathcal{R}$  is a computable realizer over a computable layered knowledge structure with knowledge space  $\mathbb{S}$ , then for all finite sound state  $X \in \mathbb{S}_{fin}$  the algorithm **Find Sound Zero** always terminates and computes a sound and finite zero of  $r$ .*

We observe that the use of the “timing” variable  $n$  is essential to the proof of termination. Indeed it encodes, though in a very crude way, the logical dependency of hypothesis from other hypotheses: when the latter are falsified then the algorithm deletes all the subsequent hypothesis since they might depend on the falsified ones. This has the effect to forbid adding to the knowledge state infinitely many hypothesis, which is at the basis of the termination proof.

The algorithm and the proof we have presented suffer of some limitations. The less severe one is the limitation of  $\text{lev}$  to the ordinal 2: in [5] we have extended the construction to knowledge structures of any finite level. A further limitation is the (non deterministic) choice of a single answer from  $r(X)$  at each step, which forces a sequential strategy instead of a parallel one. A more efficient choice would be a consistent non-empty subset  $U$  included in  $r(X)$ , whenever there is some. Also the idea of using (linear) time to avoid infinite choices of the hypothesis is not the optimal solution to the problem of removing the hypothesis that logically depend on the falsified ones. We leave the improvement of the algorithm to further work.

## 6. CONCLUDING REMARKS AND FURTHER WORK

We have defined the notions of state of knowledge and of state topology. We have then redefined in the more general setting of non-monotonic learning, the concepts of individual

(here called “solution”) and of interactive realizer that we treated elsewhere, proving the completeness of learnability w.r.t. validity, which is the counterpart of classical truth in the present setting.

The definitions and results obtained are aimed at the development of a full theory of learning strategies and of their convergence properties, which is work in progress. We also observe that the solution and the realizer illustrated in the examples of §4 are crude simplifications of the learning strategy implicit in the example of §1, which is capable of using the counterexamples in a more ingenious and efficient way. The investigation of the interpretation of classical proofs in terms of learning strategies is a natural further step, extending the work we have done in the monotonic case.

Since learning strategies working with finite approximations are effective (and indeed we have shown that finite and sound knowledge states exist and suffice), a question of efficiency of the algorithms one extracts from proofs with our method is naturally there, together with the analysis of suitable data structures representing time and logical dependancies, which are essential to complete the present approach.

## REFERENCES

- [1] Federico Aschieri and Stefano Berardi. Interactive Learning-Based Realizability for Heyting Arithmetic with  $\text{EM}_1$ . *Logical Methods in Computer Science*, 6(3), 2010.
- [2] Stefano Berardi. Classical logic as limit completion. *Mathematical Structures in Computer Science*, 15(1):167–200, 2005.
- [3] Stefano Berardi and Ugo de’ Liguoro. Toward the interpretation of non-constructive reasoning as non-monotonic learning. *Information and Computation*, 207(1):63–81, 2009.
- [4] Stefano Berardi and Ugo de’ Liguoro. Interactive realizers. A new approach to program extraction from non constructive proofs. *ACM Transactions on Computational Logic*, 13(2):11:1–11:21, 2012.
- [5] Stefano Berardi and Ugo de’ Liguoro. Non-monotonic pre-fix points and Learning. In David Baelde and Arnaud Carayol, editors, *Proceedings of FICS 2013*, volume 126 of *EPTCS*, pages 1–10, 2013.
- [6] Stefano Berardi and Ugo de’ Liguoro. A Calculus of Realizers for  $\text{EM}_1$  Arithmetic (Extended Abstract). In *CSL*, volume 5213 of *LNCS*, pages 215–229, 2008.
- [7] Stefano Berardi and Silvia Steila. Ramsey for pairs as a classical principle in intuitionistic arithmetic. <http://www.di.unito.it/~stefano/Berardi-Steila-RamseyAndLLPO.pdf>, 2013.
- [8] Giovanni Birollo. *Interactive Realizability, Monads and Witness Extraction*. PhD thesis, Università di Torino, 2013. <http://www.di.unito.it/~stefano/Birollo-PhDThesis.pdf>.
- [9] Thierry Coquand. A semantics of evidence for classical arithmetic. In Gordon Plotkin Gérard Huet and Claire Jones, editors, *Proceedings of the Second Workshop on Logical Frameworks*, pages 87–99, <http://www.lfcs.inf.ed.ac.uk/research/types-bra/proc/>, 1991.
- [10] Thierry Coquand. A semantics of evidence for classical arithmetic. *J. Symb. Log.*, 60:325–337, 1995.
- [11] Daniel Fridlender. Highman’s lemma in theory. In Eduardo Giménez and Christine Paulin-Mohring, editors, *Types for Proofs and Programs, International Workshop TYPES’96, Aussois, France, December 15-19, 1996, Selected Papers*, volume 1512 of *LNCS*, pages 112–133, 1998.
- [12] Gerhard Gierz, Karl Heinrich Hofmann, Klaus Keimel, Jimmie D. Lawson, Michael Mislove, and Dana S. Scott. *Continuous Lattices and Domains*, volume 93 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2003.
- [13] Jean-Yves Girard. *Proofs and Types*. Cambridge University Press, Cambridge, England, 1989. Translated and with appendices by Paul Taylor and Yves Lafont.
- [14] E. Mark Gold. Limiting recursion. *J. Symb. Log.*, 30:28–48, 1965.
- [15] E. Mark Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.
- [16] Timothy G. Griffin. The formulae-as-types notion of control. In *Conf. Record 17th Annual ACM Symp. on Principles of Programming Languages, POPL’90, San Francisco, CA, USA, 17–19 Jan 1990*, pages 47–57. ACM Press, New York, 1990.

- [17] Susumu Hayashi. Mathematics based on incremental learning, excluded middle and inductive inference. *Theor. Comp. Sci.*, 350:125–139, 2006.
- [18] Jean-Louis Krivine. Realizability in classical logic. In *Interactive Models of Computation and Program Behavior*, Panoramas et Synthèses, 2009.
- [19] Alexandre Miquel. Existential witness extraction in classical realizability and via a negative translation. *Logical Methods in Computer Science*, 7(2), 2011.
- [20] Jan von Plato. A Constructive Approach to Sylvester’s Conjecture. *J. UCS*, 11(12):2165–2178, 2005.